Universidad Politécnica de Madrid

Escuela Técnica Superior de

Ingeniería y Sistemas de

Telecomunicación

Context modelling for natural Human Computer Interaction applications in e-health

Author

Xin Li

Tutor

Ph. D. José Fernan Martinez Ortega

Ph. D. Martina Eckert

November 2014

MÁSTER EN INGENIERÍA DE SISTEMAS Y SERVICIOS

PARA LA SOCIEDAD DE LA INFORMACIÓN

Trabajo Fin de Máster		
Título	Context modelling for natural Human Computer Interaction applications in e-health	
Autor	Xin Li	
Tutor	José Fernán Martinez-Ortega Martina Eckert	VºBº.
Ponente		
Tribunal		
Presidente	Lourdes López Santidrián	
Secretario	Matías Garrido González	
Vocal	César Benavente Peces	
Fecha de lectura		
Calificación		

El secretario:

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my tutors Dr. José-Fernán Martínez Ortega and Dr. Martina Eckert for their extraordinary and selfless supervision and help during the whole master project. During the research, I was impressed by their patience and precious personality. They are more than professors and tutors acting as my friends as well. I have to admit that the thesis cannot be accomplished without their contributions.

My family deserves my deepest thanks for their enormous love to me. My parents, they always provide supports either financially or emotionally. Even though now we are in different countries and have seven hours time difference, they always comfort me in time whenever I feel sad or desperate and make me to firmly believe that I will never be alone because they are always there caring for me.

Finally, I want to appreciate the help from my colleagues. They are kind-hearted and much more knowledgeable than me. Their help and care for my research means a lot for me. Besides, everyone is so nice and friendly to me, working in the perfect atmosphere is a joy.

Abstract

The conception of IoT (Internet of Things) is accepted as the future tendency of Internet among academia and industry. It will enable people and things to be connected at anytime and anyplace, with anything and anyone. IoT has been proposed to be applied into many areas such as Healthcare, Transportation，Logistics, and Smart environment etc. However, this thesis emphasizes on the home healthcare area as it is the potential healthcare model to solve many problems such as the limited medical resources, the increasing demands for healthcare from elderly and chronic patients which the traditional model is not capable of. A remarkable change in IoT in semantic oriented vision is that vast sensors or devices are involved which could generate enormous data. Methods to manage the data including acquiring, interpreting, processing and storing data need to be implemented. Apart from this, other abilities that IoT is not capable of are concluded, namely, interoperation, context awareness and security & privacy. Context awareness is an emerging technology to manage and take advantage of context to enable any type of system to provide personalized services. The aim of this thesis is to explore ways to facilitate context awareness in IoT.

In order to realize this objective, a preliminary research is carried out in this thesis. The most basic premise to realize context awareness is to collect, model, understand, reason and make use of context. A complete literature review for the existing context modelling and context reasoning techniques is conducted. The conclusion is that the ontology-based context modelling and ontology-based context reasoning are the most promising and efficient techniques to manage context. In order to fuse ontology into IoT, a specific ontology-based context awareness framework is proposed for IoT applications. In general, the framework is composed of eight components which are hardware, UI (User Interface), Context modelling, Context fusion, Context reasoning, Context repository, Security unit and Context dissemination. Moreover, on the basis of TOVE (Toronto Virtual Enterprise), a formal ontology developing methodology is proposed and illustrated which consists of four stages: Specification & Conceptualization, Competency Formulation, Implementation and Validation & Documentation. In addition, a home healthcare scenario is elaborated by listing its

well-defined functionalities. Aiming at representing this specific scenario, the proposed ontology developing methodology is applied and the ontology-based model is developed in a free and open-source ontology editor called Protégé. Finally, the accuracy and completeness of the proposed ontology are validated to show that this proposed ontology is able to accurately represent the scenario of interest.

Table of contents

Table of figures

Table of charts

Table of equation

1 Introduction

1.1. Motivation and contributions

IoT is the expansion of current internet services which endows every object including people and things with the capability of understanding, thinking and talking. Its ultimate objective is to make things smart and interact with each other through internet and it is accepted as an ideal solution to enhance and make changes on Internet to accommodate the coming future. IoT is supposed to have the potential to change the world, just as how the internet did, maybe even more. There are many reasons to support of the application of IoT. One major benefit brought by IoT is "to allow people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any service" [1]. In fact, IoT is still at the primary stage and needs more efforts. People can interpret IoT with respect to their own interests and needs, such as Things Oriented Vision, Internet Oriented Vision and Semantic Oriented Vision [2]. Especially in Semantic Oriented Vision, the fact is that vast sensors or devices will be involved in the system and massive data will be generated, possibly redundant. The arising issue is to find a way of managing, understanding, processing and storing all the raw data. It is also worth noting that so far the development of IoT is not mature in terms of the performance of interoperation, context awareness and security & privacy. Among the aforementioned capabilities, context awareness is an emerging concept and technology to provide relevant information and/or services to the user, where the relevancy depends on the user's task and is found by making use of context. It is necessary to understand, use and store the context since enormous data is involved in IoT in order to provide more personalized services for customers. Therefore, it is valuable to dig into how to realize context awareness in IoT. The objective of this thesis is to explore ways to integrate context awareness into IoT. The premise of facilitating context awareness is to collect, model, understand and use context. This thesis makes a comprehensive literature review of the existing context modelling and reasoning techniques and proposes the ontology-based approach is the most promising method. The core of the ontology-based approach is to make use of ontology to represent and construct the interested domain and write the knowledge in languages such as OWL. More specifically, ontology creates a common vocabulary and allows developers to formalize a knowledge domain by means of concepts, attributes to characterize them and relations between the concepts. One of the major advantages of ontologies is their reusability. Besides, different ontology can interoperate with each other. Aiming at realizing the objective, a preliminary research is conducted in this thesis.

Overall, the major contributions of this thesis are as follows:

- A literature review for IoT is summarized and the gap between current development of IoT and the desired state is pointed out with an emphasis on context awareness.

- A thorough summary of the existing context modelling and reasoning techniques is made based on current literature. Ontology is regarded as the most promising technique to fulfill both context modelling and reasoning.
- An ontology-based context awareness framework for IoT is proposed based on some existing frameworks in literature.
- Adapted from a popular enterprise ontology development methodology, a formal approach for developing ontologies is proposed to provide detailed guidance about how to create an ontology.
- An ontology model is developed based on the proposed framework for a specific IoT scenario, more precisely, it refers to as the home healthcare system. In addition, validation for the ontology is done to show that the proposed ontology concretely represents the application of interest.

1.2. Organization of thesis

The rest of the thesis is organized as follows:

Chapter 2: A state of the art in IoT is presented. The historical development of IoT is introduced and the structure of IoT is provided. Major areas which IoT is applied to are summarized. The advantages of IoT are analyzed and several unsolved problems including interoperability, context awareness and security & privacy are pointed out. More precisely, to fuse context awareness into IoT is the major focus in this thesis.

Chapter 3: Fundamental knowledge about context and context awareness is provided in this chapter. Basic concepts such as context and context awareness are explained in detail. In addition, five methods to classify context are extracted from literature.

Chapter 4: A context process is elaborated in this chapter. A brief survey of context life cycle is made. Each phase of context process is introduced in detail. More specifically, the existing techniques used in each phase are introduced.

Chapter 5: Two major contributions are made in this chapter. Firstly, a brief investigation through available context awareness frameworks is summarized. Afterwards, a new proposal for ontology based context awareness framework for IoT is presented with detailed explanations for important modules.

Chapter 6: The research interest in this chapter is put on a highly demanded topic which is home healthcare. A specific home healthcare scenario is described by listing its main functions. Besides, a formal approach to create an ontology is proposed. Based on the proposed framework in chapter 5, an ontology aiming at representing the scenario is created and explained in detail.

Chapter 7: Major contributions in this thesis are summarized and future work is discussed as well.

2 The Internet of Things Paradigm

2.1. The Evolution of Internet

In fact, Internet has just about 60 years' development history since it came out to the public's sights in late 1960s. It has rather humble origins when it initiated. In 1969, four supercomputing centers for military research were linked by an experimental network called ARPAnet developed by the Department of Defense Advanced Research Projects Agency (ARPA) which makes the rudiment of Internet. TCP/IP stack was introduced into the network in the early 1980s, afterwards the commercial usage of Internet began. In 1990, Tim Berners-Lee invented the World Wide Web as a method of publishing information in a hypertext format. Internet became more popular and grew in a rapid speed as the adoption of the WWW. In [3], Web of Things (WoT) based on WWW is regarded as a part of IoT. Internet got a breakthrough with mobile devices connected. Since then, Internet has been the method lying in the first place to unit people by means of various social media like emails, Facebook, blogs etc. Experts say the Internet will continue along its phenomenal growth path despite of the current global economic crisis. The next evolution which will be witnessed is an extension of this Internet to everything around people namely Internet of Things (IoT). In figure 1, an overview of the evolution of the Internet from [3] is given.

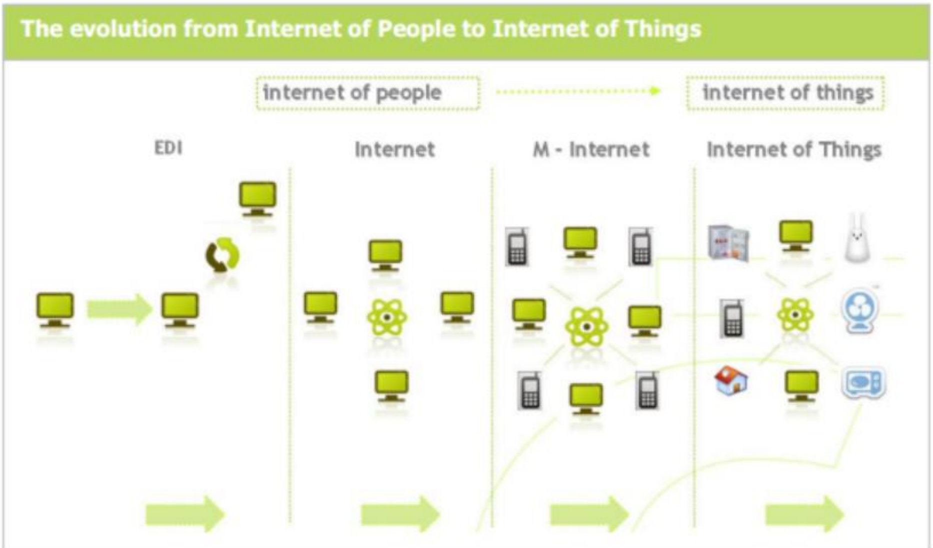

Figure 1. The Evolution of Internet to Internet of Things (IoT) [3]

2.2. Internet of Things (IoT).

Literally, the ultimate intention of IoT is to make things smart and interact with each other through the internet. It is the expansion of current internet services so as to endow each and every objects the ability of understanding, thinking and talking. Everything is enhanced to accommodate the coming future. Recently, IoT is drawing more and more attention from not only academia but also industry. As a novel concept and technique, brainstorm among researchers has been started. The interest in this area can be seen roughly from figure 2 which shows the searching results of the keyword "Internet of Things" in the database IEEE Xplore.

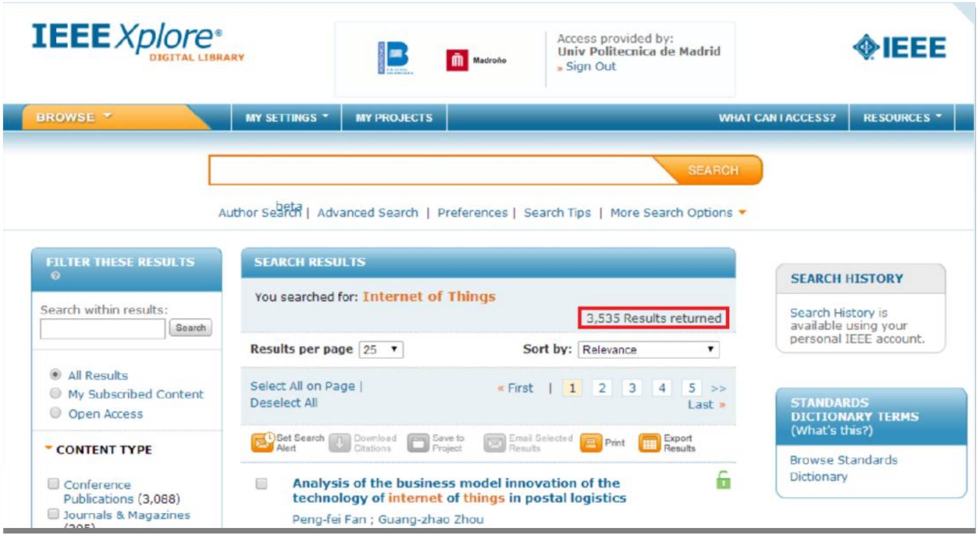

Figure 2. A search for "Internet of Things" in IEEE Xplore returns 3,535 results

As IoT is still at the first stage of development, there are no standard definitions for it until now. The term "IoT" was firstly proposed in [4] by Kevin Ashton in 1999. The nomenclature was mentioned in a presentation "*The Internet of Things has the potential to change the world, just as the internet did. Maybe even more.*"

Aiming at researching IoT in depth, the Auto-ID center was founded to focus on EPC-IoT in 1999 in Massachusetts Institute of Technology (MIT). In 2005, the International Telecommunication Union (ITU) officially introduced IoT and in the meanwhile four important technologies of IoT were proposed in World Summit on the Information Society (WSIS) conference. The first international conference of IoT was hold at Zurich in 2008 that means IoT finally shows its significance and entered on the scene. In the following a set of definitions for IoT which have been highly accepted and used will be listed in order to give a clearer understanding of IoT:

- "Being embedding short-range mobile transceivers into a wide array of additional gadgets and everyday items, enabling new forms of communication between people and things, and between things themselves" was proposed in WSIS (World Summit on the Information Society) in 2005.
- "... describe a number of technologies and research disciplines that enable the internet to reach out into the real world of physical objects..." was proposed in Internet of Things 2008 Conference.
- "Things having identities and virtual personalities operating in smart spaces using intelligent interfaces to connect and communicate within social, environmental and user contexts" was proposed in [5].
- "The semantic origin of the expression is composed by two words and concepts: Internet and Thing, where Internet can be defined as the world-wide network of interconnected computer networks, based on a standard communication protocol, the Internet suite (TCP/IP), while Thing is an object not precisely identifiable Therefore, semantically Internet of Things means a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols" [6].
- "The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and any service" [1].

The aforementioned definitions come from different visions. Generally, the last explanation as illustrated in figure 3, encapsulating a broad vision, is the most precise, reasonable and widely accepted by public.

Figure 3. The definition of IoT in [1]. The IoT allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and any service

2.3. Observation for IoT from different views

As it is widely agreed by researchers, IoT is still staying in a nascent stage which everyone interprets it with respect to their own interests and needs. In general, three particular visions including things oriented version, internet oriented version and semantic oriented version, from which people develop IoT, are discussed (figure 4).

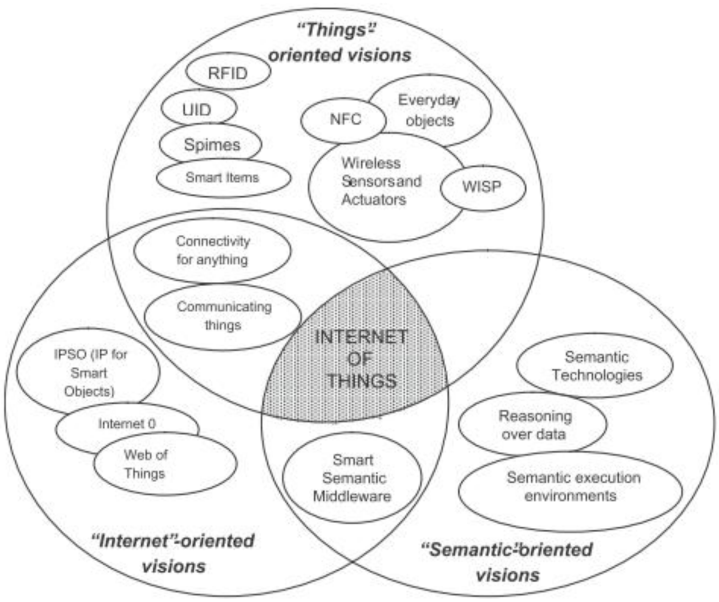

Figure 4. Three main visions of IoT from [2]

2.3.1. Things oriented version

SNs (Sensor Networks) reveal their significance in IoT due to the fact that they provide powerful support for IoT in many aspects. Sensors and actuators are used to collect various types of information. Combined with pervasive technologies using RFID, many functionalities are realized, such as detecting and tracking objects etc. The philosophy behind it is using specifications of an Electronic Product Code (EPC) to identify any object involved in the system. RFID becomes the main force driving this vision because of its maturity, low cost and strong support from the business community. Things oriented vision will depend on sensor based networks as well as RFID-based sensor network which integrates RFID and sensing techniques to obtain connectivity for anything.

2.3.2. Internet oriented version

At a first glance at the term "Internet of Things", two keywords are notable, namely "Things" and "Internet". From the perspective of Internet oriented vision, network or networking is the core of IoT. United Nations predicted the advent of IoT during the meeting in 2005 in Tunis and proposed this concept. Any object can be connected to the internet obeying the same IP protocols. For now, the IP stack as a light-weight protocol has already connected a variety of communicating devices. This reveals the possibilities of making IoT reality. IEEE 802.15 [7] is proposed to be added into IP architecture to promote the large-scale deployment of IoT. Those proposals mentioned above pave the way to the development of IoT in a rapid speed.

2.3.3. Semantic oriented vision

It is worth noticing the fact vast sensors or devices will be involved in the system accompanied with massive data, possibly redundant. The arising issue is finding a way to manage, understand, process and store all the raw data. In this context, semantic technologies are presented as a potential way to represent, store, interconnect and search information. Suitable modelling solutions can be exploited to organize all the raw data in an understandable manner and also make it possible to infer more meaningful information.

Semantic technologies play an important role in accommodating IoT requirements and scalable storing and communication infrastructure.

The so called Web of Things in [8] is supposed to be the further vision of IoT. In this vision of IoT, all objects containing an embedded device or computer will be connected and integrated in the web by reusing web standards.

2.4. The essential component of IoT: Sensor Networks

It is valuable to have a study on a SN which is taken as the most essential component of IoT. Indeed, managing a network of sensors is not a new concept, at least it emerged earlier than IoT. In the following, a brief introduction for SN will be presented.

The core of a SN is sensor nodes. A SN comprises amounts of sensors which can be either homogeneous or heterogeneous. Communication among each sensor node can be realized wired or wireless while in reality wireless sensor technology takes the dominating place. Besides different technologies and protocols can be exploited to connect different SNs, one striking approach is made through internet. There are several popular wireless technologies applied to build wireless sensor networks: Wireless Personal Area Network (WPAN), Wireless Local Area Network (WLAN), Wireless Metropolitan Area Network (WMAN), Wireless Wide Area Network (WWAN). Two protocols are widely used in sensor networks: non-IP based and IP-based protocols.

Figure 5 from [8] shows the internal components of a sensor node.

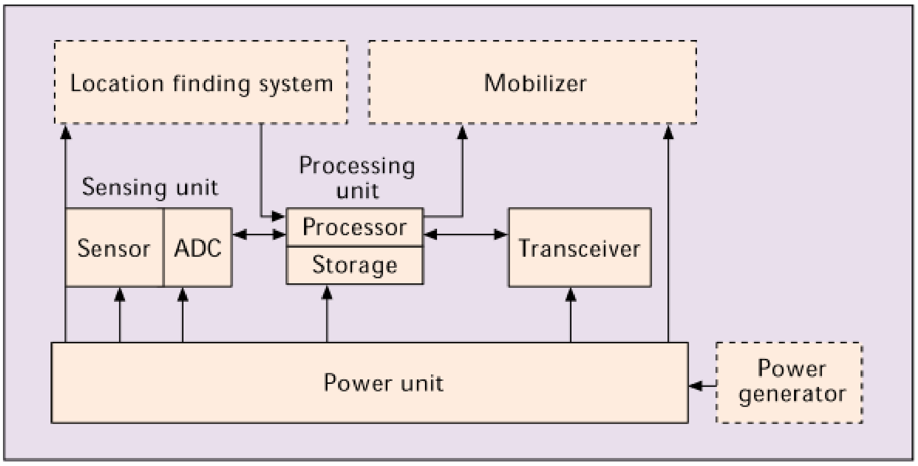

Figure 5. The components of a sensor node

As depicted in figure 5, four basic components constitute a sensor node: a sensing unit, a processing unit, a transceiver unit and a power unit. Additional components such as a location finding system, power generator and mobilizer can be adopted into it according to the specific application. An ADC (Analog-to-Digital converter) does the job of converting the analog signal detected by sensors based on the observed phenomenon to digital signals, then the digital signals are fed into the processing unit and go through the following process. The whole sensor node can connect to the network because of the transceiver. Most of SNs need to be aware of location because it remains an important parameter for network routing techniques and sensing tasks. Sometimes a mobilizer may be introduced into the node when sensors are required to carry out the assigned tasks8.

In [8], the existing architectures in SNs are classified into three types. The first and the simplest is a flat architecture. The scheme behind it is transferring data from static sensor nodes to the sink node in a multi-hop method. Some complexity is added into the second kind of architecture named two-layered architecture with more static and mobile sink nodes deployed to collect data from sensor nodes. The last architecture (three-layered architecture) is referred to as the most complex and powerful architecture which allows the connection of multiple SNs through internet and is highly recommended for IoT.

A typical structure of a SN is presented in figure 6. This SN incorporates the most common components. As can be seen from the picture, the workflow is roughly like these procedures as followed: low-end and high-end sensor nodes detect physical data and send them to static and mobile sink nodes, next the data is transferred to low-end computational devices for further process. Finally, data comes to the cloud and a thorough process and usage of data will begin.

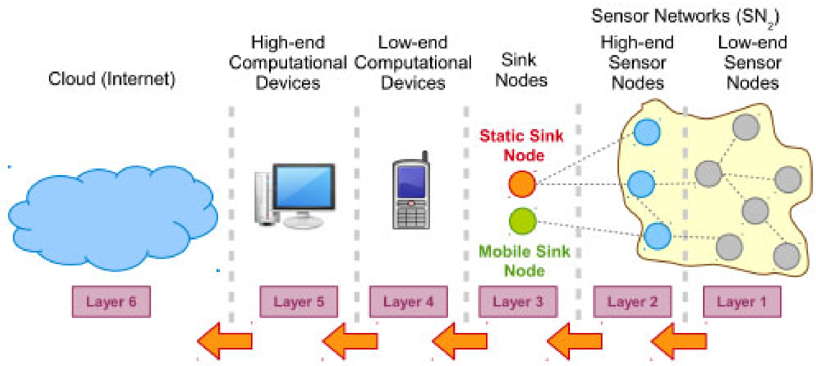

Figure 6. Layered structure of a SN [8]

In this structure, six layers are identified by the capabilities of the devices. The concept of capability can be understood in many ways, it concludes the ability of processing, memory, communication and energy capacity. From the lowest to the highest layer, more and more capabilities and power are endowed to the devices. An ideal system should be capable of distinguishing the difference of capability and providing data management. A clear and reasonable allocation of data processing should be made carefully. For instance, the first few layers just need to do some simple process for data because of the limit of energy and processing power. In this way, the burden of data processing is shifted to the higher layer and in the meanwhile data in communication is reduced. It is evident that data management is a key issue in IoT, the way of data management is always needing to be improved.

2.5. Relationship between SN and IoT

After studying the IoT and SN concepts, in this section the relationship between SN and IoT will be described. The SN is highlighted as the essential component of IoT because of its importance in many aspects. Figure 4 already shows how the SN is integrated in the IoT as an indispensable component. Further, the relationship between SN and IoT can be clearly seen from figure 7. Sensors and actuators are basic hardware in IoT. They are the origins generating all data for further process. Then the system can make decisions on the basis of the data. Currently, many publications can be found about how to integrate SN with IoT. However, the difference between SN and IoT is blurred. To clarify it, a list of differences between SN and IoT are presented in the following and some of the features of SN and IoT to analyze their relationship are elaborated.

- A SN is composed of hardware (like sensors and actuators), firmware and software whereas the IoT contains some of the components of the SN and additional software components like middleware systems, frameworks and APIs etc.. The software layer is installed between computation devices and the cloud to connect hardware with application layer.
- By tracing their origins, it is found that a SN is created to realize a specific application or purpose. At a very early age, most of SNs were used for monitoring purposes instead of actuation. To fulfill a certain purpose, a special set of sensors which is relevant to an application is deployed into the system. In contrast, IoT is referred to as a general purpose SN. It can support different applications for different needs. Therefore, high amounts of sensors have to be employed in the IoT in order to provide sufficient data for various application domains. Middleware solutions, frameworks and APIs also need to be designed in a general way for providing generic services and support. To meet all these requirements, IoT should be intelligent enough to know the changing need of each application and provide information correspondingly.
- A SN can work as an independent network without IoT while IoT cannot exist without a SN. The reason behind this is that the SN comprises the major part of the hardware infrastructure in IoT. A SN provides this support through accessing to sensors and actuators. In fact, several other technologies can complement the IoT infrastructure such as wireless ad-hoc networks. However, the lack of scalability and flexibility limits the usage of wireless ad-hoc networks in IoT. Figure 7 clearly demonstrates the relationship between a SN and the IoT.

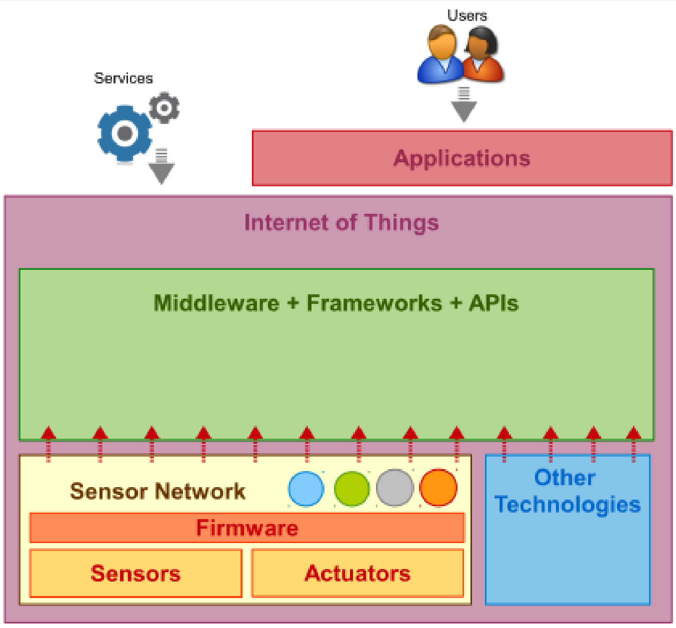

Figure 7. Relationship between a SN and the IoT [10]

2.6. Middleware support for IoT

The definition in [9] can provide a good explanation about what exactly middleware is: "Middleware is a software layer that stands between the networked operating system and the application and provides well known reusable solutions to frequently encountered problems like heterogeneity, interoperability, security, dependability." In other words, middleware acts as a software platform to provide abstraction getting from things to applications and supplying various services. It is an active topic of research in IoT, the main reasons are listed as follows:

- Middleware can act as a bond to unit all the heterogeneous components.
- A common standard will be difficult to establish for diverse hardware belonging to different domains in IoT.
- A abstraction/adaptation layer is strongly required for a variety of applications.
- Middleware provides support for physical layer communication by the adoption of API (Application Programming Interface) and requested services to specific applications, in this way, the details of diversity are hidden.

Due to its significance in IoT, middleware is researched in many publications proposing solutions. [10] presents a thorough survey of the leading solutions for middleware from several aspects like device management, interoperation, platform portability, context awareness and security & privacy. Table 1 shows the classification of those middleware solutions based on different features.

IoT Middleware	**Features of Middleware**				
	Device Management	Interoperation	Platform Portability	Context Awareness	Security and Privacy
HYDRA	√	√	√	√	√
ISMB	√	×	√	×	×
ASPIRE	√	×	√	×	×
UBIWARE	√	×	√	√	×
UBISOAP	√	√	√	×	×
UBIROAD	√	√	√	√	√
GSN	√	×	√	×	√
SMEPP	√	×	√	√	√
SOCRADES	√	√	√	×	√
SIRENA	√	√	√	×	√
WHEREX	√	√	√	×	×

Table 1. IoT middleware comparisons from [10]

As depicted from the table, almost every middleware solution can meet the fundamental requirements like device management and platform portability. Other functionalities such as interoperation, context awareness and security & privacy remain as obstacles needing to be tackled.

2.7. Services and applications of IoT

IoT can be applied to any system to fulfill any purpose as a generic sensor network. It enables any object with the sense of "thinking" and sufficient intelligence to make its own choices. Overall, based on the up-to-date development of IoT, an overview of popular domains which IoT is applied into is conducted.

2.7.1. Healthcare area

Healthcare domain is always a significant issue affecting all people's life. With the mergence of IoT, more breakthroughs have been obtained during the past years. IoT is proposed to be applied to any place related to healthcare like hospital or home.

The basic function which IoT can provide is tracking. The essence of tracking is a function that recognizes an object (or a person) and follows it. Patients can be tracked continuously so professionals can be aware of their daily activities. The deployment of wireless sensors or other devices also can monitor the status of a patient's health, e.g. his temperature, humidity, skin temperature, heart rate just to name a few. This information is vital for doctors to make prescriptions or other measures precisely.

Another benefit is identification and authentication. All data related to patients can form a patient's medical history or electronic record. The maintenance is critical to avoid any modification or intercept. For achieving this goal, only people who are identified and authorized can be allowed to get access to the electronic record. In this way, the safety of patients can be guaranteed.

Moreover, IoT gives healthcare domain more intelligence, e.g. doctors can obtain suggestions for prescription generated by the system. As a result, medical professionals are able to shift part of their obligations to an intelligent software or system.

With the adoption of IoT in the healthcare domain, patients can enjoy more customized services and care. The unbalance between limited medical resources and increasing need from patients can be bridged.

2.7.2. Transportation and logistics

The Intelligent Transportation System (ITS) is another popular application with IoT, e.g. vehicles do not need to stop to pay the fee when they drive across a toll station because they have RFID tags inside and RFID technique can automatically identify the location and collect toll information. Driving on the road, vehicles can receive important information about the road condition, the status of traffic jam, the occupation of a specific road and available routes etc. from the transportation control center. Drivers can easily avoid traffic jam and choose the best way to destination based on the real time information.

IoT is also proposed to be applied into logistics which is a hot domain in the world. The key point is to track goods and check their quality. Clients can track the up-to-date information about their goods by just checking the tracking system online. The logistics company also can use RFID to locate the objects to check if they are shipped as the plan. Besides, when the goods happen to be stolen, IoT can help to locate and find the lost goods.

2.7.3. Smart environment

A smart environment is a quite general concept. It refers to any place with intelligence and its principal aim is to improve life of human beings. The smart home is a notable application which brings so many benefits for people's life, as fundamental environment parameters are monitored and adjusted according to the changing environment and people's preference. For example, the temperature inside the house is measured continuously, when the current temperature value is higher than an exact threshold, without the human's intervention, an air conditioner can be turned on to cool the air. Similarly, other devices have the abilities to adapt their behavior to the changing environment. On the other hand, all devices or electronic equipment can be controlled by users remotely. People can make an order to turn on air conditioner before they come back home. In the agriculture domain, IoT plays a critical role which integrates smart technology with human's knowledge. Sensors embedded in the ground send kinds of soil parameters to a decision maker to make a better plan for watering and fertilizing. In this way, crops can get the necessary water or fertilizer on time. The waste of water and fertilizer can be avoided while the production of food grains can be increased.

The applications of IoT are pretty much wider and not limited to those areas mentioned above. More domains should be explored to take advantage of IoT.

2.8. Research gap in IoT

IoT still can be improved based on its unsolved problems as identified previously. As can be seen from table 1, not all mentioned existing IoT middleware solutions can meet all main features required by IoT, but they all solve device management, so the way to connect sensors to middleware is provided by these solutions. In addition, these solutions can be transplanted in other platforms and in other words it refers to platform portability. It can be found from table 1 that interoperation, context awareness and security & privacy are remaining as obstacles needing to be conquered. Apart from those, there are still more open issues which need more research efforts. For instance, standardization is a missing component in IoT. It is worth pointing out that so far IoT is lack of mature standardizations. In addition, because of the huge number of nodes in IoT, effective addressing policies are required to accommodate the increasing need. In this paper, the issue of context awareness in IoT is trying to be solved. In general, even though there are several middleware solutions with context awareness functionality, they are still having drawbacks that limit the satisfactory from IoT. In the next chapter basic concepts related to context awareness are introduced.

3 Context and context awareness

3.1. Context

It is a challenging issue to define the term of "context" since it varies with the changing environment and the involved persons. Many researchers have made efforts to make definitions for context in their literature from different aspects, but according to the work from Dey et al. [11], it holds the opinion that the majority of the existing definitions are not accurate and comprehensive. Dey made a survey of evaluating the existing definitions and highlighted the weakness of these definitions. Here several definitions for context which are widely accepted by public are presented:

- **Schilit** and **Theimer** [12] pointed out that "*context is the set of location, identities of nearby people and objects and changes to those objects*."
- **Brown**, **Bovey and Chen** [13] referred to context as "*location, identities of the people around the user, the time of day, season, temperature and so forth* in a similar way."
- **Ryan**, **Pascoe** and **Morse** [14] thought context is "*the combination of the user's location, environment, identity and time*."
- **Franklin** and **Flachsbart** [15] described context as a specific situation "*What is happening at this moment*".
- **Ward**, **Jones** and **Hopper** [16] viewed context as the state of the application's surroundings.
- **Hull**, **Neaves** and **Bedford-Roberts** [17] insisted that "*context is just the aspects of a current situation*."
- **Chen** shared his understanding for context in [18] which is: "*extending to model the activities and tasks that are taking place in a location*".
- **Henricksen** [19] said that "the context of a task is the set of circumstances surrounding it that are potentially of relevance to its completion".
- **Oxford dictionary** defines context as "*the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood*."

Taking a holistic view on the aforementioned definitions, it is not difficult to find that some of them only focus on a particular application or just enumerate the entities for context. In other words, they lack of generality and standardization.

Additional improvements are made based on these definitions. Schilit et.al explained context like this: where the user is, who the user is with and what resources are nearby. The constantly changing environment is regarded as context. Afterwards, researchers dedicated themselves to seek the operational definitions for context in literature [20] [21].

Dey's explanation for context is the most widely accepted operational definition to encompass all the previous definitions. It is general and accurate enough as it contains sufficient expert knowledge.

Context defined by **Dey** is: " *any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups and computational and physical objects.*"

In [21], the authors made a detailed survey on the existing research on context definitions, after analyzing each of them, they listed a comparison results for them and obtained their own understandings for context. The authors proposed a new operational definition for context which is generated from the components of context's point of view. In their opinion, context should be understood by making clear its elements. Figure 8 can explain the authors' idea for context explicitly and concretely.

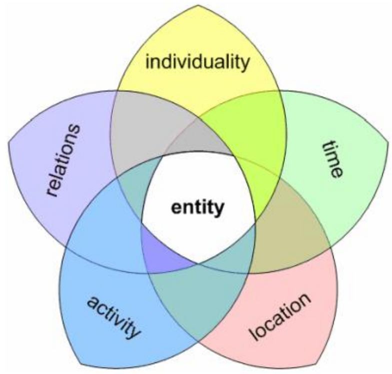

Figure 8. The definition of context in [21]

As can be seen from the figure, the idea in [21] is to split the term of context into several sub-items and apply relationships to connect different sub-items, so what they want to express in [21] is that context is composed of all these concepts. This approach helps to analyze its components step by step and individually.

As a conclusion, the definition for context defined by Dey is quite reasonable, and it is highly agreed and accepted in the reminder of this paper.

3.2. Context classification

As it is revealed in section 3.1, context is any information that occurs in an application or environment which includes either external factors or the users and application themselves. The context is a notion which could contain any kind of information. The following questions should be answered.

Does some of these amounts of information have something in common ?

Is it possible for part of them to be attributed to a same property or range?

Are there some objective laws for the context?

How to understand, manage and sort out the context in an efficient way?

The way to cope with those questions is to classify the context. Given the diversity of context information, it is meaningful to spend time on it because with a clear classification people can easily comprehend the context. Aiming at solving those doubts above, a comprehensive survey on the existing methods to sort out context is conducted. Five means of categorizing context from different aspects are introduced in the following including 5W1H, Physical/Virtual Context classification, Static/Dynamic Context classification, Sensed/Combined/Inferred/Learned Context classification and Conceptual/Operational Context Classification.

3.2.1. 5W1H (Why, Who, What, Where, When and How)

In [22], context is sorted out by the category of 5W1H, more specifically, it is Why, Who, What, Where, When and How. Any kind of context can be attributed to one of these elements. The classification originates from the point that focuses on what is happening in the specific scenario or environment. Each type has its own specific role and function for the whole context. More details of this method can be seen in table 2.

Category	Duty	Description	Examples
WHO	Role	Basic user information/ Role can be any agent that involved in the context like person, organization etc.	Name, gender, age, preferences, habits, characteristics
WHEN	Time	A record of time like year:month:day:hour:minute:second	Time stamp, time of day, season, holidays, anniversary
WHERE	Location	Locate the place of objects	Coordinate (x, y), place, region, environment
WHAT	Status	Relevant objects	Applications, services, commands (application dependent), software agents
HOW	Action	Ongoing processes happening in the context	Signals from sensors, e.g. current activity of the user, information from cameras
WHY	Goal	The intentions or objectives of the ongoing processes	Stress, emotion, gestures, objectives, future events from schedules

Table 2. Classification for context in 5W1H discipline

This table demonstrates that six types make up the complete context. The factor to differentiate and classify all the context information is to identify their function and role in the environment. For example, context like name, gender, age, preferences, habits and characteristics or other similar information are regarded as type "Who" because they provide necessary information related to the basic background of users or objects. The common feature is to describe everything relevant to the users and make stakeholders know more about users. Similarly, other contexts are divided to other types because of their properties and features. Making use of this classification, contexts are well-organized and easily understood regardless of its huge amount and diversity.

3.2.2. Physical/Virtual Context classification

This method simply classifies context into two categories, they are physical context and virtual context, respectively. [23] specifies the detailed classification rules and reasons for this method which establishes the basis of the kind of classification method. However, the authors in [23] insist the opinion that physical context is equivalent to location information which is not so correct and comprehensive. This explanation which limits physical context to location contradicts to the fact that physical context should be a set of various types of context. In this light, the updated explanation for context should be like figure 9.

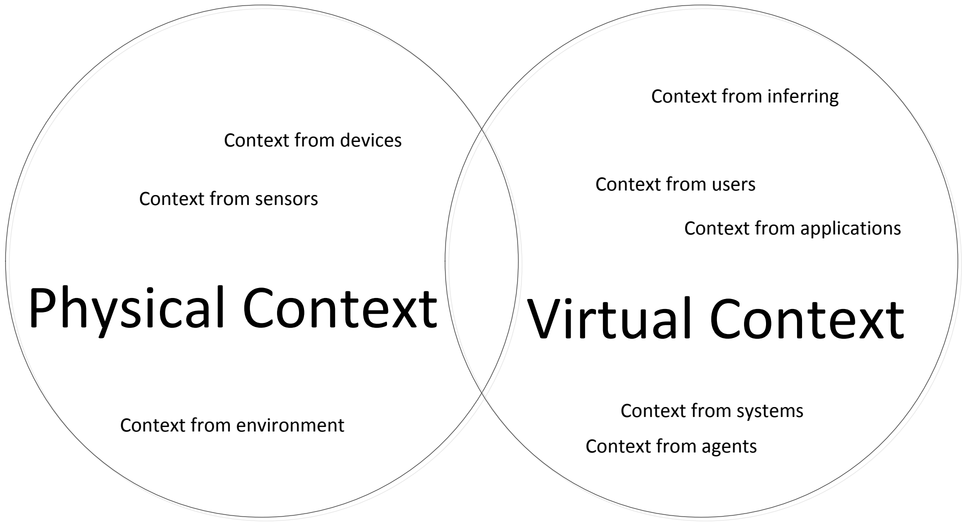

Figure 9. Physical/ Virtual Context classification

Contexts are distinguished based on their sources or origination. Figure 9 illustrates the proposed viewpoint on context classification which actually is an improvement work based on [23].

- **Physical context**. Physical context is obtained from sensing devices like sensors, camera etc. The data is detected, gathered, aggregated from the environment and represents the varying environment. Normally it can be evaluated by numeric numbers which change as the varying environment. It is a big-build project if you want to count and list all physical context. Typical physical contexts are weather, temperature, humidity, acceleration, vibration, presence, decibel levels, velocity, light intensity, wind strength, wind speed, air pressure, location, movement, sound, touch, index of human body quality like heart rate, blood pressure, skin temperature etc. Physical context is regarded as the basic and essential context which is important and widely used in every context-aware application.
- **Virtual context**. It refers to as logic context in some publications. Anyway, either virtual context or logic context, represents the same range of context which comes from different sources compared with physical context. It can be deduced directly from physical context or other virtual context. Besides, users can provide the data for virtual context. For example, through human-computer interface, users can make a set of initialization and choice to import their personal information. Commonly used

virtual contexts are tasks, goals, personal preference, medical history and other useful personal information.

It can be concluded that the sum of physical and virtual context covers all the context, which means that this kind of classification is quite clear, efficient and easy to understand.

3.2.3. Static/Dynamic Context classification

Another classification for context is the characterization as static or dynamic context according to its attributes.

- **Static context**. Static context describes the kind of context information which is eternally invariant despite of the change of time or other intervention from environment. In other words, the value representing static context is constant. It can be used in the whole process once it is provided because it never change. Because of the strict constraints to meet this attribute, there are not so many contexts which can be categorized into this type. For example, birthday, age and gender are obviously divided into this type since they are fixed once the user is born.
- **Dynamic context**. It changes as the change of time or the change of other factors. It cannot keep the same status at most time. Due to the nature of IoT, it can be deduced that the majority of contexts in IoT are dynamic contexts. As IoT is typically characterized by frequent changes, so the persistence of dynamic context information can be highly variable.

3.2.4. Sensed/Combined/Inferred/Learned Context Classification

Another method to classify context is proposed in [24] by separating it into four different categories as described in the following:

- **Sensed-context.** It is similar to physical context. It stands for those contexts based on sensing devices. It can also refer to as raw context directly obtained from sensing devices without any process or calculation. It is the basis for other context types.

- **Combined-context.** It is not collected directly from the environment. Applying a set of simple calculations for the sensed-context, the result is combined-context. For instance, the thermometer can monitor the varying temperature in the environment with providing a specific value. The unit of this value is normally Celsius degree. So how to get the correct value in Fahrenheit degree in the application with a clear requirement for unit? A simple formula to convert Celsius (abbreviated to C) to Fahrenheit (abbreviated to F) can be applied as equation 1.The result Fahrenheit degree is combined-context. So the combined context seems to be the fabrication of sensed-context as a metaphor.

$$F = {}^{9}/_{5} * C + 32 \qquad (1)$$

- **Inferred-context**. Here, inferred-context regarded as high level context information is inferred from sensed-context by applying some rules. Indeed, it is the harvest obtained by people's efforts to dig more valuable context information from the sensed-context.
- **Learned-context**. This kind of context is obtained by exploiting learning algorithms like DT (Decision Tree) or NN (Neural Network). By learning, analyzing the history of other contexts like sensed-context or combined-context, learned-context can be generated by a decision making process.

3.2.5. Conceptual/Operational Context classification

Abowed et al. [25] firstly put forward a leading mechanism to categorize context. In their opinion, the term of context can be split into two types: primary type and secondary type. Location, identity, time and activity are identified as primary context. Secondary context is defined as the extension or process for primary context. [26] points out the shortcoming of this mechanism that when dealing with a specific context value is that it is hard to distinguish its type clearly, as the boundary between primary context and secondary context is blurry. Afterwards, the authors in [26] expressed their own understanding for this mechanism and a new classification as figure 10 is generated as an evolution of the previous mechanism.

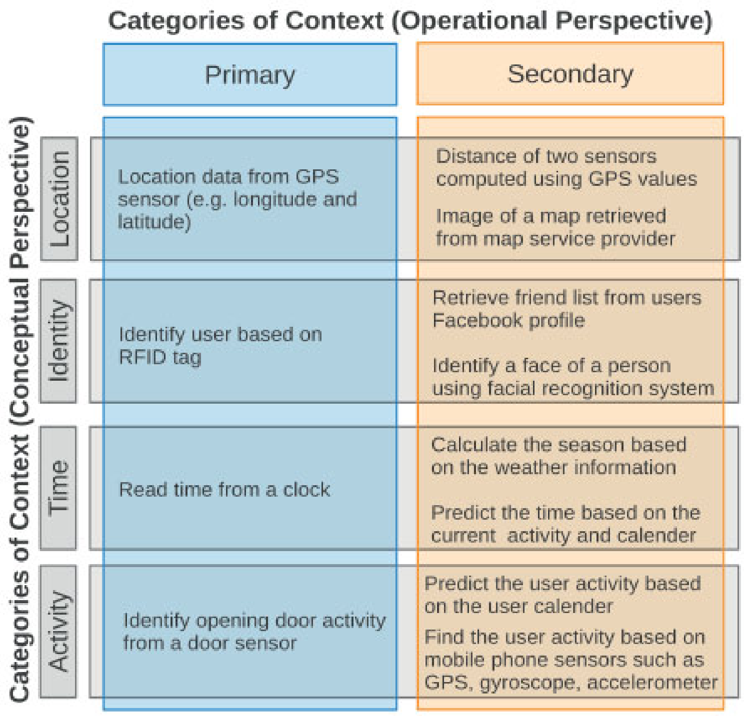

Figure 10. Context categorization into two different perspectives from [26]

The evolved mechanism considerably solves the puzzle to identify a same context value in different situations. Context information can be obtained from different sources. The source is a key factor to determine the context type. Keeping this in mind, the authors designed a new method to identify context types in two perspectives: conceptual and operational. As can be seen from the figure 10, even though different context values are related to the same entity like location from the conceptual perspective, they can be treated as primary or secondary context because of their different origins. The mechanism is a breakthrough in defining context because it combines two different perspectives.

- **Primary context**. As mentioned previously, primary context contains any context information or data retrieved from sensing hardware like sensors and camera etc. without any process, fusion or calculation. In other words, primary context is pure or raw context without any intervention from human. It keeps the same as it originally is.
- **Secondary context**. Secondary context is the extension of part of primary context. Secondary context cannot be retrieved without the existence of primary context. For example, a person's identity can be collected by recognizing his RFID tag. Based on

his identity, a set of additional information is available like use's preference, social life (friends, spouse etc.) which in fact is secondary context.

The major contribution of this mechanism is to identify context in not only the operational but also conceptual perspective. In the operational perspective, data acquisition techniques are taken into consideration and the context is categorized on the basis of study for acquisition techniques. With the fusion of these two methods, the categorization can be more precise and reasonable.

3.3. Context features

Context information has special features compared with normal information. In this subsection, a sketchy study of context characteristics will be conducted. Several major features are elaborated in the following.

Context information can be huge. Any kind of data which could be useful for the realization of a system with context awareness can be regarded as context information. It is evident that the amount of context information could be enormous. Due to this, computing burden is a concern which needs to be paid attention to. Efficient computing methods are needed to handle those enormous data. A possible way is to borrow computing techniques from ubiquitous computing area. Another concern arises which is it is needful to find a way to manage the enormous information.

Context information is imperfect. It cannot be guaranteed that all information obtained to build a context is correct and valuable. There are many reasons leading to this imperfection. First of all, the sensing devices possibly dismatch the real need which means the selection of sensing devices is unable to detect the specified context information. As a result, the sensing context data can not represent context concretely. Secondly, sometimes even with the correct sensing device it can fail in getting the correct data because of limited functions and performance. This imperfection cannot be solved fundamentally because all what people can do is to improve the accuracy rate of context and the performance of sensing devices.

Context information exhibits a range of temporal characteristics [27]. In the previous section, a categorization of context information which classifies context into static and dynamic types has already been introduced. Beyond this background, it is known that the majority of context information is dynamic as the environment in IoT is always changing and the frequency of variation of context information is very fast. So normally context information is valid only temporally, for example, to automatically control the temperature of a house the first step is to get the real time temperature. A thermometer is deployed inside the house to detect the changing temperature. Only current temperature is dominant to the decision of control, but the gathered value can be just valid for a short period of time until it is replaced by the next value. Many types of context information have the same feature.

Context information has many alternatives [27]. To acquire a specific type of context information, one can choose from a wide amount of sensors. Different values or data in different formats can represent the same content reflecting the change of the current state which gives users a lot of flexibility to select the most suitable one. For example, to find a person's location, GPS is capable of supplying the concrete raw coordinates of the person. In the meanwhile, a camera deployed in the living room can also prove the presence of the person. Additionally, an infrared ray can also detect that the person's activity is taking place in the living room. This feature is the major advantage of context information. Users can choose the most suitable context information for a specific need.

Context information is sensed through sensors or sensor networks. The present situation is that most of context information is sensed through sensors or SNs. Sensors and SNs play a very important role in supplying context information since they are the prime source compared with other media. SNs are treated as a database so as to become one of the promising data management solutions.

Context is sensed by small and constrained devices. Looking at the features of sensors, it can be found that currently most sensors used for detecting context information are small and cheap. On one hand, these features are advantages from economic perspective because small size and cheap devices are convenient to convey and install. On the other hand such devices have limited computing power. Additionally, the limited battery capacity is another concern.

So energy management should be designed carefully in order to decrease the data burden and increase the work life which points out the future work.

3.4. Definitions of context awareness

The term of context awareness, also called sentient, was first proposed by Schilit and Theimer [28] in 1994. A context awareness system has the ability to adapt to the changing environment and provide services accordingly, like the definition from Dey:

"*A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.* [29]"

All kinds of context data can be utilized to let the system know more about users, environments and become more intelligent. This definition for context awareness is highlighted as the most general and accurate explanation. Thanks to its generality, this definition can be fit in any context-aware application. The historic evolution of the definition is introduced in the following part.

Hull et al. [30] and Pascoe et al. [31] proposed their understanding for context awareness as the ability of computing devices to detect and sense, interpret and respond to aspects of a user's local environment and the computing devices themselves. Dey put forward a narrow definition for context awareness which is limited to the human-computer interface [32]. Then in [33], Dey refined his definition by introducing the notion of adaptation which regards context awareness as the work leading to the automation of a system by making use of user's context. Salber et al. [34] thought that if a system is context aware, then it can provide the maximum flexibility of service based on real time context. Brown [35] defined that context aware applications can automatically provide information or take actions according to the user's present context and need. A slight difference lies in [36], Ryan defined that context awareness focuses on the specific methods to achieve it, more specifically, in his opinion if an application has the ability to monitor input from sensing devices and choose the suitable context according to user's need or interests then it can be attributed to be a context-aware application.

However, most of the aforementioned definitions are too specific so when coping with another situation they cannot be used. Seeking the similarity from these definitions, it is revealed that context awareness is synonymous to some extent with some other terminologies like adaptive, reactive, situated, context-sensitive, responsive and environment-oriented. In this work, it shares the same understanding as Dey's which describes the term of context awareness in a more general and abstract way.

The term of context awareness has a very close relation with the term of context, they are always bonded together inextricably. A context-aware application is allowed to use context and adapt to the environment. The acquisition of context is the premise to achieve the ability of context awareness for an application. And context awareness is the goal and desired state for acquiring and processing context in an application.

3.5. Context awareness features

In this subsection, the main work is to look deeply into the features of context awareness from an IoT perspective. The authors in [36] elaborated the characteristics of context awareness from an IoT perspective.

- **Presentation**. The principle goal of a context-aware application is to automatically decide which kind of service or information should be presented to the user. To achieve this goal, context is used by the application as a critical criterion. Presentation means to show what the user really needs and display the action determined by the system. For example, assuming a smart shopping mall, a customer steps into the first floor where he needs to know the detailed information about the layout of the building and commodity. So the context awareness system can retrieve the necessary information from the inner system of shopping mall. If the user is equipped with smart phone or tablet, he can request any information he is interested in from the phone or tablet. Additionally, the phone or tablet can display extra popular information like goods on discount, the highly recommended restaurants, the acclaimed movies etc. The essence of presentation based on varying context guarantees the realization of IoT because it is helpful to provide any service at anytime, anyplace with anything for anyone.

- **Execution**. A major characteristic for a context awareness application is that it can automatically execute missions and provide services accordingly without the intervention of human beings. The system knows when it is the right time to execute the task or run the facilities deployed in the system. The system can totally understand what people are thinking and supply the best and right services. For example, assuming the circumstance is in a context awareness office. On weekdays, the facilities in the office will run correspondingly to people's preferences. A set of actions will be executed before people come to the office, e.g. :
    1) The air conditioner can turn on by itself.
    2) The air conditioner can adjust itself to the most comfortable temperature.
    3) The light can turn on several minutes earlier before people enter in the office.
    4) The computer can power on and start in advance.
    5) The humidifier can start working.

  All in all, the appliances are able to be ready to welcome the people's coming. They have the sense to change their states according to people's needs and habits. The benefits brought by this feature are considerable which enhance the system with the ability of context awareness.

- **Tagging**. As mentioned in chapter 2 amounts of sensors will be deployed in IoT in the future. The data gathered from sensors will be the resource to extract useful information namely context and the huge amount of data is a growing concern. In most situations, a single kind of context data can not reflect the real changes of the circumstance. Different types of context data should be integrated to fully understand the situation. Under this background, context will be tagged with different sensor data which will be processed later [36]. The annotation is very important for the further process and understanding.

3.6. Context awareness levels

In [37], the authors elaborated three different levels of interactivity between the computing devices and users.

- **Personalization**. The context-aware application can be customized or personalized according to users' preferences. It points out that personalization of desktop applications is the hot research topic currently [37]. However, the potential area for personalized applications can be much larger. The tailoring software can solve those issues brought by the diversity and dynamics of personalized applications. For example, the humidifier deployed in a house can keep the same humidity as the one retrieved from user's preference instead of the default comfortable humidity. When the user turns on the TV, it will switch to the user's favorite channel and play it autonomously. Personalization makes the services oriented to users more smoothly.
- **Passive context awareness**. The system continuously checks the changes of the environment and provides a set of choices for users. The system cannot execute any operation without the permission of users. They just present the related choices based on the varying circumstance for users. The user is the master who determines which kind of service is allowed to execute. Finally, the selected operation is executed and presented to the user. Compared with active context awareness, the system does not have the same rights to automatically provide the corresponding services. The choice of the user is a critical factor affecting the operation of the system. For example, when a smart phone detects there is an updated software available on the internet, a message will be sent to user to notify this information about software updating information. After having the awareness of this alert, the user can make his decision based on these three options:
    1) Update this software to the newest version right now.
    2) Ignore this message now but show it one day later.
    3) Do not update this software this time and do not show this information any more.

- **Active context awareness**. Active context awareness is somewhat the opposite of passive context awareness. It is enhanced with more autonomies. The process to let the users choose their options is neglected. The system is able to provide services for users automatically without their intervention. Using the same example in passive context awareness section and changing it to active context awareness system, the smart phone can update autonomously every time it detects new version without notifying users of this change.

4 Context process

After introducing the concepts of context & context awareness and its importance in IoT, the focus is forwarded to the next step: how to achieve the ability of context awareness in IoT with the awareness of the detailed context life cycle.

In this section, the content will be organized as follows:

Firstly, a brief review of the context life cycle is conducted on the basis of the newest survey [38].

Secondly, introduction for each phase is presented in detail as the same sequence of the context life cycle. Aiming at each phase, existing techniques are demonstrated as the supplement of the way to enhance IoT with context awareness.

At last, a simple conclusion is given to overview the usage of this section in high-level and point out the subsequent work.

4.1. Context life cycle

Up to this point, only the basic definitions of the concepts of context and context awareness have been studied to provide the fundamental knowledge for a deeper research on this area. The essence of context is data collected from various sources. In fact, all the data are pointless until they are used. The details related to how the data is acquired, how the data is understood, how the data is shared, how the data is consumed and how the data is stored etc. are still indistinct. The whole process mentioned previously is referred to as a data life cycle. The true meaning of a data life cycle is to list and study the flow of data from the obtainment to the usage. Data is metaphorized into one kind of creature endowed with life. So data has its cycle that specifies the loop from the time it is generated to the moment it becomes useless.

Because of their close relationship with data, context also has a typical life cycle similarly. Context shares so much resemblance with data that it is easy to find many disciplines during the process of its usage and based on these disciplines indentify the typical procedures for its life cycle.

Substantial efforts have been put into this area and different understandings for context life cycle came out. At the very beginning, the understanding of context awareness is limited to several kinds of applications, more specifically, like desktop, web and mobile applications. However, in [39], authors have a more open thought for context awareness from a novel perspective. Context awareness was proposed as a service: Context-as-a-Service (CXaaS). As a consequence, context management shows more and more importance in software system especially in IoT. [39] categorizes data life cycles into two types: Enterprise Lifecycle Approaches (ELA) and Context Lifecycle Approaches (CLA). CXaaS will enable users to focus more on the functionalities for their specific applications rather than the development of context management components.

- **Enterprise Lifecycle Approaches (ELA)**. As the name says, the data lifecycles are dedicated for enterprise applications. Context is not the goal of these data management. ELA is to identify the lifecycle of sensitive data in enterprise application, the big difference between ELA and CLA derives from the property of data involved: most data in ELA is static. For example, an employee's profile including gender, age, height, blood type etc. do not vary. Moreover, the benefits are that there are enough mature and well-established standards and regulations for these data management from industry area. To some extent, the ELA are suitable for all general data. Thanks to the strong support from the industry area like some of software vendors, such as Oracle and Microsoft, ELA has experienced the development of prosperity.
- **Context Life Approaches (CLA)**. The research object of CLA is limited to context data. They are more relevant to the interest of this thesis and they inspire people to have their own understandings for context lifecycle. Compared with ELA, CLA is staying at the primary stage which is lack of standard strategies and should be tested. And many CLA have not gone through the test procedure yet.

As mentioned previously, with many efforts put into this area, currently, researchers have different understandings for context lifecycle and propose various lifecycle approaches. In the following, some widely accepted approaches are explained and illustrated as an example, they are much more.

- **Information Lifecycle Management (ILM)** [40]. It consists of the lifecycle, the tools, practices and methods surrounding the lifecycle. In [40], the authors define ILM as "the policies, processes, practices, services and tools used to align the business value of information with the most appropriate and cost-effective infrastructure starting at the information is created until its final disposition". The specific steps can be seen from figure 11.

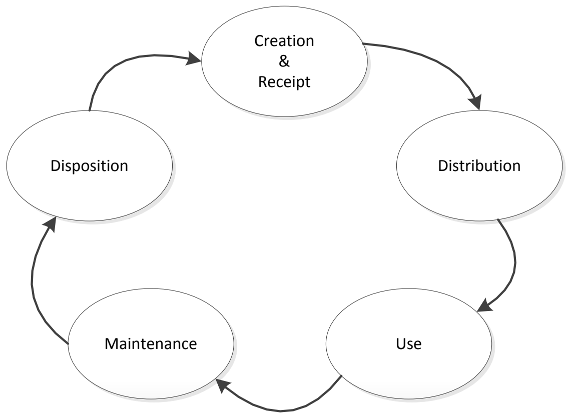

Figure 11. Information Lifecycle Management

- **Enterprise Content Management (ECM)** [41]. Similar as ILM, ECM is made of the lifecycle, tools, practices and methods. ECM is "the strategies, methods and tools used to capture, manage, store, preserve, and deliver content and documents related to organizational processes" [41]. Figure 12 depicts the concrete procedures for ECM. It can be found that ECM focuses more on those steps before data dissemination.

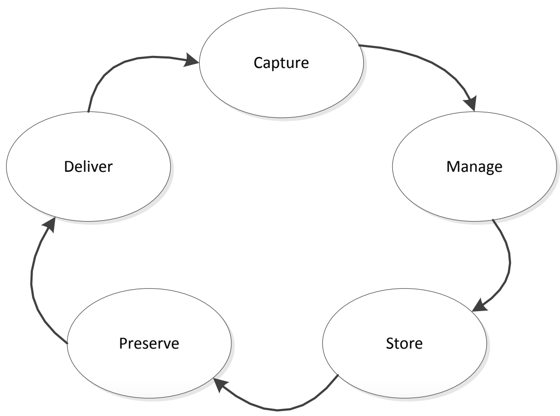

Figure 12. Enterprise Content Management

- Hayden [42] proposed a data lifecycle management strategy which is the fusion of ILM and ECM. It firstly takes the factor of users into consideration. Users are capable of altering data in an unauthorized manner during the lifecycle with less impact to the data accuracy. The whole lifecycle contains ten steps, more specifically, they are collection, relevance, classification, handing and storage, transmission and transportation, manipulation and conversion/alteration, release, backup, retention and destruction. More details can be found in figure 13.

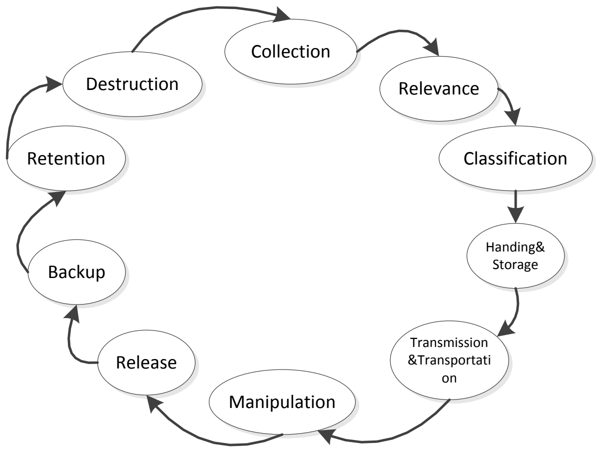

Figure 13. Lifecycle proposed in [42]

- **Intelligence Cycle** [43]. This data lifecycle approach is an advanced method in comparison with the aforementioned methods because it includes a step called *feedback* which analyzes the results of data use which can be an important factor to influence the data collection. So the management software can collect data according to the real need analyzed from feedback. This method is an intelligent one as it can provide data changeably based on the feedback information. In general, five stages constitute the intelligence cycle, they are collection, processing, analysis, publication and feedback as illustrated in figure 14.

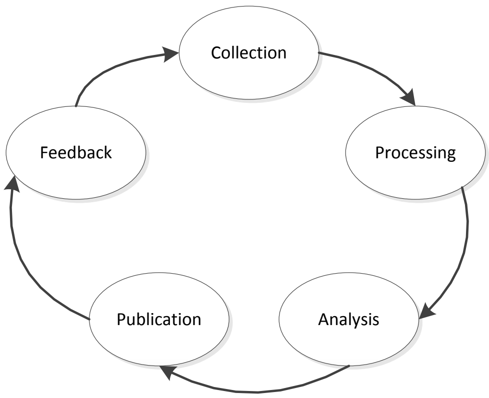

Figure 14. Intelligence cycle [43]

- **OODA Loop** [44]. It is also called Boyed Control Loop. This strategy derives from the military field which is intended to solve the decision making. Generally, the OODA sketch is made up of four components: observe, orient, decide and act. OODA loop is defined in [44] as "the OODA loop sketch and related insights represent an evolving, open-ended, far from equilibrium process of self-organization, emergence and natural selection." Figure 15 presents the whole picture of OODA.

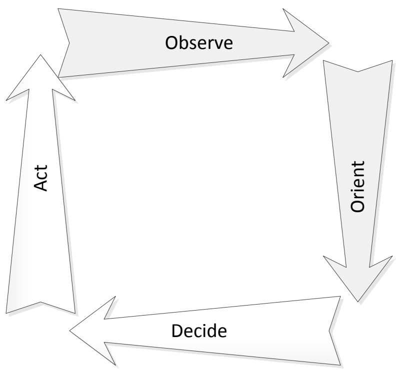

Figure 15. OODA loop

All presented data lifecycle approaches elaborated above belong to Enterprise Lifecycle Approaches (ELA). As members of ELA, they share the same properties as ELA. Next, several typical CLA will be introduced briefly.

- Chantzara and Anagnostou in [45] put forward a data lifecycle which classifies the whole procedure of consuming data into four separate stages. Three types of actors like Context Providers, Context Broker and Service Providers are responsible for executing the missions during the lifecycle. More details are shown in figure 16.

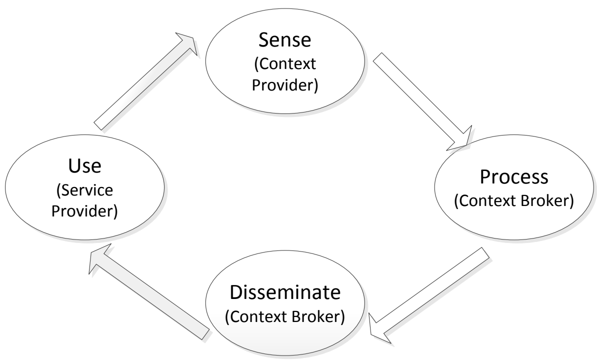

Figure 16.Context lifecycle in [45]

- Fersha et al. [46] introduces a more complex context lifecycle approach compared with the one in [45]. The approach figures out the specific process from acquiring low level context data to the last stage of consuming context data. More specifically, five steps formulate the approach and they are Sensing, Transformation, Representation, Rule Base and Actuation. The sketch can be seen from figure 17.

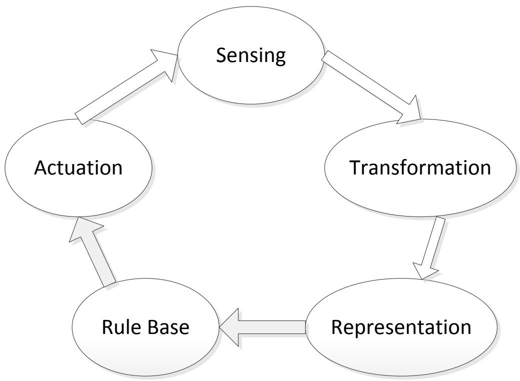

Figure 17. Context lifecycle in [46]

- WCXMS (**Web-based Context Management Service**) Lifecycle [47]. It is the extension of lifecycle in [45]. It is a quite complete context lifecycle which covers the whole procedure from the time of generating context to the end of consuming context. The design idea is inspired by the aforementioned lifecycles. For example, the idea of Context Acquisition stage is borrowed from MOSQUITO project. Figure 18 shows the whole picture of this lifecycle.

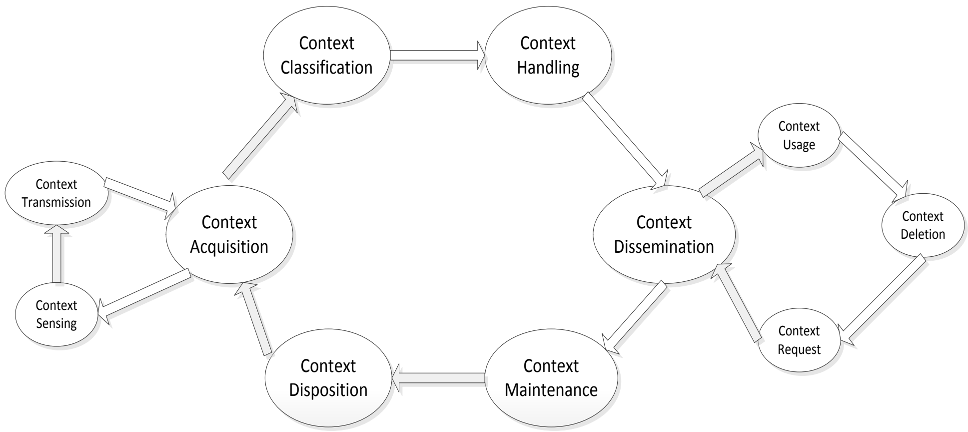

Figure 18. WCXMS lifecycle [47]

- Baldauf et al. [48] indentified the context lifecycle as five stages: sensors, raw data retrieval, reprocessing, storage and application.
- Perera et al. [38] came up with a simple context lifecycle as figure 19 which divides the context lifecycle into four stages: Context Acquisition, Context Modelling, Context Reasoning and Context Dissemination.

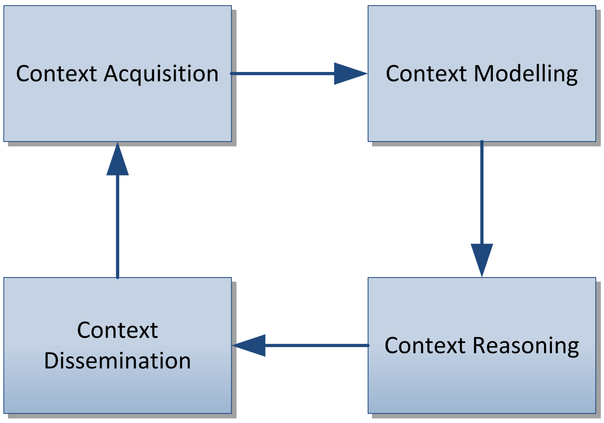

Figure 19. Context lifecycle designed in this thesis

Table 3 analyzes the advantages and disadvantages of all the approaches mentioned previously.

Data Lifecycle Approach	General Category		Advantages	Disadvantages
	ELA	CLA		
ILM	√		More focus on the steps after data distribution, data sources are not limited to sensors, most data involved is static, simplicity.	Data representation and formalization is missing, the usage of data is in low level, no feedback.
ECM	√		Concern with the storage step, most data involved is static simplicity.	Lack of detailed process for data like modelling and reasoning, the usage of data is in low level, no feedback.
Hayden's	√		Users are allowed to manipulate data, procedures are complete.	Too complex, the policy for controlling user's alteration for data is missing
Intelligence Cycle	√		Simplicity, adaptive with the adoption of feedback, closed control loop.	Lack of detailed process for data like modelling and reasoning, data storage is missing.
OODA	√		Simplicity with only four steps, focus more on decision making.	Lack of detailed process for data like modelling and reasoning, no feedback.
Chantazzara and Anagnostou's		√	Simplicity.	Data source is limited to sensors, the process step is too ambiguous.
Fersha's		√	Simplicity, the usage of context data can be in high level.	Lack of data storage, data source is limited to sensors.
WCXMS		√	The context data is complete, user is allowed to manipulate data.	Too complex, applicable for web-based applications.
Baldauf's		√	Simplicity.	Data source is limited to sensors, the first two steps are similar, lack of feedback.
Perera's		√	Simplicity, data sources are not limited to sensors, the usage of context data can be in high level, focus more on processing context data.	Lack of data storage and feedback.

Table 3. Summary of discussed approaches

As can be seen from the table, there is no perfect data lifecycle approach. Among all, those approaches which are attributed to CLA are more interesting and suitable to deal with context data. In particular, Perera's approach is the simplest one but contains all essential components. In this work, the viewpoint is shared with Perera as figure 19 because of its simplicity and emphasis on the steps before the data dissemination.

The seamless cooperation of these four phases is building the entire lifecycle. In the phase of context acquisition, amounts of data are collected from the environment. The data can be retrieved from various sources, like directly gathered from physical world or collected from users. The context modelling phase formulates all the data from the last phase into an understandable manner. In this way, differently formatted data is structured in a unified format in order to make it meaningful. High level context information is generated in the context reasoning phase and ultimately all context data is employed.

In the rest of this chapter, each phase will be examined in detail and the techniques in each phase will be introduced. As the focus of the thesis is on context modelling and context reasoning, those phases will be explained with most emphasis.

4.2. Context acquisition

Being the first phase, the context acquisition module is responsible for obtaining all context data as exactly as possible from diverse sources. This phase shows its significance during the whole procedure of realizing context awareness in an application because it is a prerequisite. The value, the number, the accuracy etc. of context data is subject to this phase. In chapter 3, the popular classifications for context made by current literature have been outlined. In the next step, the techniques and devices used for acquiring context are examined.

4.2.1. Techniques to acquire context

The fact that context is categorized into physical and virtual context is widely accepted among academic researchers. Upon this background, the techniques for obtaining different types of context can be diverse. The following study of context acquisition is conducted based on the context sources, in other words, where the context data is obtained from.

How to acquire physical context?

According to the definition of physical context it can be found that physical context works with data obtained from sensing devices. The selection of the sensing devices is determined on the basis of the specific requirements in different applications or systems. It is worth noting that sensors are the most widely employed appliance for obtaining context from the surrounding. Generally, they are converters which monitor the changing aspect of the physical environment and convert the changes into another modality (mostly electronic signals). With the help of sensors, context consumers are capable of knowing the real time change of a particular property which they cannot get directly by observing or touching. Generally every sensor is sensitive to one special phenomenon. However, for detecting the similar aspects of the environment, more than one sensor should be available. As an example, table 4 lists some usually used sensors.

Name	Raw data	Captured contexts
Audio sensors	Sound	Noise, music, decibel levels
Video sensors	Figures, images, sound	Emotions, presence, gesture, movement, behavior
Motion sensors	Optical changes, acoustic changes	Presence, single or multiple users
Pressure sensors	Pressure	Pressure, occupation, hand gestures, touch, movement
Light sensors	Resistance, electrically charged signals	Ambient light, intensity of indoor brightness
Accelerometers	Proper acceleration, coordinate acceleration	Motion, vibration, physical state
Bluetooth	Short distances	Location
Infrared	Invisible radiant energy	Location, presence
RFID	Tags, electromagnetic induction	Location, activity, situation, profile
GPS	Location, time information	Location, direction, movement
Temperature sensors	Electrical resistance	Temperature
Oxygen sensors	Proportion of oxygen	The amount of oxygen
Speedometer	Speed	The instantaneous speed
Carbon dioxide sensors	Proportion of carbon dioxide	Density of carbon dioxide
Seismometer	Seismic waves, motions of ground	Crustal movement, vibration
Humistor/Leaf sensor	Water loss, water deficit stress, humidity	Humidity
Weather sensors	Weather changes	Weather
Hydrogen sensor	Proportion of hydrogen	Density of hydrogen
Smoke detector	Smoke	Density of smoke, fire
Voltage detector	Voltage	Voltage
Rain sensor	Rainfall	Rain

Table 4. Selected widely used sensors

Table 4 does not represent an exhaustive list for available sensors. In literature, much more extensive summaries can be found. For example, the Wikipedia webpage [49], seems to have the most complete list for existing sensors.

Although sensors have high importance and cannot be replaced by other instruments in context acquisition, they are not perfect. It is valuable to make a simple analysis for sensors' shortcomings in order to know more aspects of sensors. The hardware field is experiencing a fast development recently. The result is that the diversity of sensors grows while their price is falling, the size of the sensors is getting smaller accompanied with less power-consumption. However, the disadvantages behind the improvements are that the quality and stability of sensors may not be as expected. So when choosing the right sensors, as many factors as possible should be taken into account and the balance among those features should be kept.

A sensor is not the only sensing device used in most IoT applications to detect context, there are others acting as supplements which are able to capture additional properties and have the capability of computing which sensors cannot. For example, a set of sensing devices usually used in context aware applications can be seen in table 5.

Sensing Devices	Aggregation	Preprocessing
Snap2Play	Location	Pilot test
ConPhone	Location	X
CAMP	Scene	Optimizer
CACH	Environment	X
CMMM	Location	MySQL
SmartTVs	CCD camera, microphone	Hand-gesture recognition
sPhones	GPS, microphone	Speech recognition
Kinect	CCD camera, laser camera Microphone	Body gesture recognition Speech recognition
Xtion	CCD camera, laser camera Microphone	Body gesture recognition Speech recognition
iPad2	Touch pad, camera Microphone, bluetooth	Proprietary software
MyFi	Voice recognition Spatial information parser	XM satellite radio station Wireless FM transmitter

Table 5. Sensing devices usually used in context aware applications

In most situations, a single sensor or other sensing device is not capable to monitor the changing property of environment precisely. With the cooperation of those sensing devices, the sensing results are more reliable and trustable.

How to acquire virtual context?

In chapter 3, an introduction to virtual context was given and distinguished from physical context. The way to retrieve virtual context can be manifold. In general, virtual context can be either provided manually or derived from other context. More specifically, users are also an important source to provide information such as birthday, age, preference, height, weight and other personal information. Besides, the context already obtained can be used to infer more context meanings by employing certain reasoning rules. Because of the unique characteristics of IoT, virtual context can also be retrieved through a middleware infrastructure. Middleware infrastructure bridges sensor hardware and specific applications. As an example, a widely used middleware is Global Sensor Networks (GSN). Application can obtain context from GSN rather than getting from real sensor hardware directly. What's more, context server is another source to provide virtual context. In this mechanism, context can be read from some context storages like databases, RSS (Really Simple Syndication) feed, web services et al.

In summary, context acquisition is gathering context from various sources. In principle, it is desired to obtain a maximum amount of data for maximizing the possibilities to provide customized services for the application with richer context information. Additionally people can make use of many techniques which are already available and useful for acquiring context such as wireless communication, image recognition, GPS, data mining, machine learning and other related techniques.

Since the context acquisition phase is not of the main interest in this thesis, no more studies are presented on this topic. In the following, the research moves on to the main interests in this document, namely they are context modelling and context reasoning. Aiming at these two fields, popular techniques are described, respectively.

4.3. Context modelling

In the previous phase, enormous context information is obtained which are structured in different formats. However, to make use of them, the premise is to define and store context data in a machine processable form. To realize this goal, all context should be structured in an unified format so the context can be understood and shared. Under this background, a model should be developed to define, represent and process the object "context".

A typical interpretation for a context model from [50] is "*A context model identifies a concrete subset of the context that is realistically attainable from sensors, applications and users and able to be exploited in the execution of the task. The context model that is employed by a given context-aware application is usually explicitly specified by the application developer, but may evolve over time*".

Generally, context models can be categorized into two types: static and dynamic [38]. In static models, there are predefined context concepts. When dealing with a specific case, context data are collected to fill all the predefined context concepts. In summary, the context information in static models are fixed to some extent, they do not change according to different applications. For example, in [51], the authors classify context into three subfields: user context, things context and system context. And they define classes for each subfield in advance. While applying the context model to a real application, data are gathered based on the predefined classes. No more context information beyond the scope of the static models will be employed. A dynamic model is the inverse of a static model which accepts as much context information as possible. Besides it is easier to adapt to the changing environment while a static model is simple to use. Some requirements should be taken into account while modelling context information, e.g. heterogeneity, mobility, relationships, dependencies, timeless, imperfection, reasoning, usability of modelling, formalisms and efficient context provisioning.

Many surveys for popular context modelling techniques have been published recently like [38] and [52]. The description for each context modelling technique is quite detailed, however those surveys are not comprehensive because some important and new modelling techniques are not included. Based on the existent efforts, in the following, a new survey for examining the existing modelling techniques including key-value, markup, graphical, object-oriented, logic-based, multidisciplinary, domain-focused and ontology-based context modelling techniques will be presented. The fundamental scheme to summarize the available context modelling techniques is based on the data structure used for representation and exchanging.

4.3.1. Key-value context modelling

It is recognized to be the simplest model. Key-value pairs are used to enumerate attributes and context information is described as values. The context information can be written in different formats such as text files and binary files. Because of its simplicity and ease of use, it was widely employed in early research and various service frameworks. Schilit in [53] describes the limited number of location information as key value pairs. When the context information becomes vast, the key value context modelling is not suitable any more. In conclusion, it cannot meet the increasing demand of representing complex contexts and gradually less used in the late research because it is lack of scalability.

4.3.2. Markup context modelling

This technique is referred to as tagged encoding. Context information is stored within tags. In fact, this technique is the evolution of key value context modelling. Context information is structured to markup tags with attributes and content. A text document consisting of symbols and annotations represents and formats context information. These symbols and annotations originate from markup languages with semantics or without semantics. Profiles are the typical representation of this modelling technique. By making use of tags, context data can be achieved more easily and efficiently. Besides, several sophisticated validation tools provide support for validating popular markup techniques.

Two classical markup context models are User Agent Profile (UAProf) and Composite Capabilities/Preferences Profile (CC/PP). CC/PP is derived from Extensible Markup Language (XML) which is the extension of the Resource Description Framework (RDF). The limit of UAProf and CC/PP is that their hierarchical structure is pre-defined. CC/PP is proposed as a profile representation by World Wide Web Consortium (W3C). It defines a basic structure for profiles. The hierarchy of the data structure is fixed in most time such as a strict two level hierarchy: each profile is composed of a set of components and each component consists a set of attributes.

Researchers have listed several fundamental requirements for context representation which can be used to evaluate the performance of a specific context modelling technique [54]. In brief, the main demands for a context representation are structured, interchangeable, composable/decomposable, uniform, extensible and standardized.

More markup languages are developed as extensions to UAProf and CC/PP standards. Compared with UAProf and CC/PP, those approaches as the descendant of RDF inherit the basic vocabulary and procedures from them but with more abilities of adapting to dynamic context information. For example, Comprehensive Structured Context Profiles (CSCP) proposed in [54] is based on Resource Description Framework (RDF) and it addresses the shortages of CC/PP and allows for all flavors of context information. What's more, it is capable of describing users' preferences and it does not define any fixed hierarchy. The pity behind CC/PP or similar context modelling languages is that there are still some problems unsolved in many aspects such as capturing relationships, dependencies, timeless, inconsistency checking, reasoning and uncertainty removing from contexts.

4.3.3. Graphical context modelling

Generally, graphical context modelling is better than key value and markup context modelling in terms of expressive richness because it is extended by specifying relationships. In this field, three widely used model examples are Unified Modelling Language (UML), Entity Relationship Model (ERM) and Object Role Model (ORM).The UML is a standardized general-purpose modelling language which is used to model context. Different kinds of context information are written in the format specified in UML. Besides, contextual information can be clearly expressed by graphical diagrams. ERM and ORM work for designing and querying databases at the conceptual level. Several graphical modelling techniques are available at the low level such as SQL (Structured Query Language) database, MySQL database and XML. Database is able to store vast amounts of context data. But a problem is arising with the adoption of different databases, more specifically, the interoperability cannot be met. The data retrieval mechanisms of SQL are limited. Additionally, complex SQL queries are needed to meet the sophisticated context retrieval demands. In SQL, it is very difficult to create, use and execute even with some existing tools. Moreover it is also an obstacle to add, modify context information in SQL.

Looking at the conceptual level, it can be found that some variants of ERM and ORM are developed to model context. In [55], Context Modelling Language (CML) is put forward referring to one kind of the extensions of ORM. The crucial improvement brought by CML is the model is capable of dealing with temporal context information. Facts are categorized in terms of the sources and properties of various context information. In addition, CML makes its contribution to support decision making about imperfect context information. Privacy can be guaranteed in CML model because CML can make a textual notation for specifying the ownership of facts. The more attractive ability of CML is its mapping to the relational model regardless of space constraints. Nevertheless, CML is not perfect from some aspects. For instance it still does not solve the interoperability essentially. The way to trace the context information source is also a remaining issue.

On the whole, graphical context modelling is a powerful method to express conceptual context information and the relationships among them by making use of graphical diagrams. However the interoperability is still remaining as a problem which needs to be tackled and configuration must be required. In addition, standards should be established in the future.

4.3.4. Object-oriented modelling

The object-oriented modelling technique employs class hierarchies and relationships to represent context data. It incorporates encapsulation, inheritance and reusability into context expression. Instances can be allowed to access the contexts by some inheritance mechanism. Entity is the core component which is the subject of structured context information. An entity is linked to other entities and attributes which represent their related properties. The relationships among entities are treated as associations. For example, in [56] an object model for a context sensitive tourist GUIDE is developed by employing this technique. Several object modules are connected by relationships to make up the whole model as specified in figure 20.

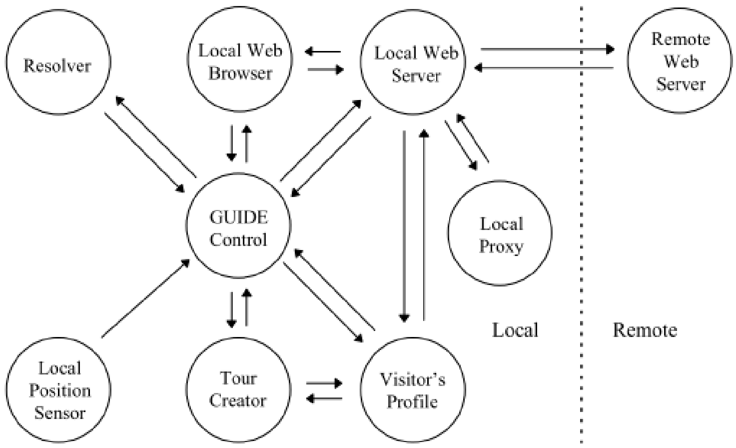

Figure 20. The GUIDE object model

Context information is structured around each object. The arrows appeared in the figure specify the relationships between different object modules. More details about concepts, relationships and organization are not given as it just presents an example to see the application of object-oriented modelling technique.

It is worth noting that this modelling technique requires the users to be aware of the whole context taxonomy before they begin to develop the model. Because of its complexity the usage of object-oriented modelling is limited to those devices with high computation ability. Similar as graphical context modelling, currently it is lack of standards. And the validation for this technique is also an unsolved problem. What's more, retrieving information is a huge challenge for this technique and reasoning is not allowed in this model. So future efforts should be focused on these fields to overcome all the aforementioned shortcomings.

4.3.5. Logic-based context modelling

In principle, facts, expressions and rules are used to express context. It allows to add, update or remove data from the current context. Rules which refer to as policies, constraints and preferences can be used in this model to extract more facts from the existing context facts. Besides, rules can be useful in other context modelling techniques like ontology-based modelling. Compared to the previously presented modelling techniques it is a big breakthrough either in expressive richness or in supporting the future reasoning process. A variety of logic-based context models have been created based on concrete applications. For example, in [57] a logic-based context model is developed. In the model, context is defined as a seven field data structure (subject, predicate, object, time, area, certainty, freshness). Context information is formalized in this sequence. It is worth noting that this context model is meaningful to check and manage context consistency. Logic-based context modelling is able to provide powerful support to the next reasoning process. The common issue also exists in this modelling technique which is lack of standards. And the way to validate this technique is still under research.

4.3.6. Multidisciplinary context modelling

Multidisciplinary is firstly proposed in [58]. Literally it can be inferred that multiple disciplines are involved and integrated in this modelling technique such as psychology, computer science and linguistics. The multidisciplinary context modelling demonstrates context from different points of view and specifies the relationships among multiple disciplines. The understanding for this modelling technique is quite complex. It incorporates the information concerning applications, users, and environment.

In general, multidisciplinary context modelling is beneficial to help people have an insight into the context process from multi-disciplines. However, because the proposal is still at the beginning, the research just stays on the conceptual level. The specific procedures how to represent context are not figured out and the real usage of this technique is rare. Moreover, interoperability remains as a problem which needs to be conquered.

4.3.7. Domain-focused context modelling

Domain-focused context modelling which refers to as the W4 (Who, What, Where, When) context model is tailored to model a application domain. [59] elaborates the mechanism of this modelling technique. Four fields tuple (Who, What, Where, When): "Someone or something (who) does/did some activity (What) in a certain place (Where) at a specific time (When)" is recognized to express any piece of context information. The component Who is to declare the subject which can be people or any object who performs the activity. What implies the detailed action or activity which is done by Who. WHERE and WHEN are auxiliary information to describe the exact place and time in relation to the fact. In other words, WHO, WHAT, WHERE and WHEN describe the fact from four different aspects. The four fields tuple is called knowledge atoms in other literature. Software agents can create the tuples for the original context data captured by data sources such as embedded devices, sensors, users etc. After that the context data structured in the uniform format W4 is stored in suitable places such as web sites or a repository. This model is very convenient for users or applications to retrieve data tuples via diverse query mechanisms. A simple example can be seen in table 6.

WHO	WHAT	WHERE	WHEN
Jack	Sleep	Bedroom, bed	07/03/2014, 23:00 h

Table 6. A simple example for W4

Table 6 shows a simple tuple to express a piece of context information:

Jack is sleeping in his bed located in bedroom at 23:00h on March 7th of 2014.

In conclusion, domain-based context modelling is quite expressive and flexible to represent context and in the meanwhile it provides various ways for users to make queries, modify and delete the context tuples. This modelling technique improves the functionalities of context aware application considerably.

4.3.8. User- centric context modelling

The essence of user-centric context modelling is a derivative of ontology-based context modelling techniques which will be explained in the next section. Most context models are analyzed from an application's perspective. However in this model, context is observed from a user's perspective and it is explored how context information is perceived by users instead of devices, services or applications [60]. 5W1H (WHO, WHEN, WHERE, WHAT, HOW and WHY) introduced in chapter 2 is explored to express the sequence of context.

This model classifies context into three different types: Preliminary Context (PC), Integrated Context (IC) and Final Context (FC). PC contains raw data from sensing devices and the features of context data. PC are kept in IC together with inferred context. FC refers to those context which are really employed into the applications: either receive or send. The same piece of context information can be structured in different formats while it is in different context type. For example, while taking *height* as the preliminary context, it can be expressed *Jack's height* as **height**= x directly. When the same context *height* exists in integrated context, it can be expressed as **height**= z-y. In this formula, z is David's height and y is the height difference between David and Jack.

A user-centric context model example extracted from [60] can be seen in figure 21.

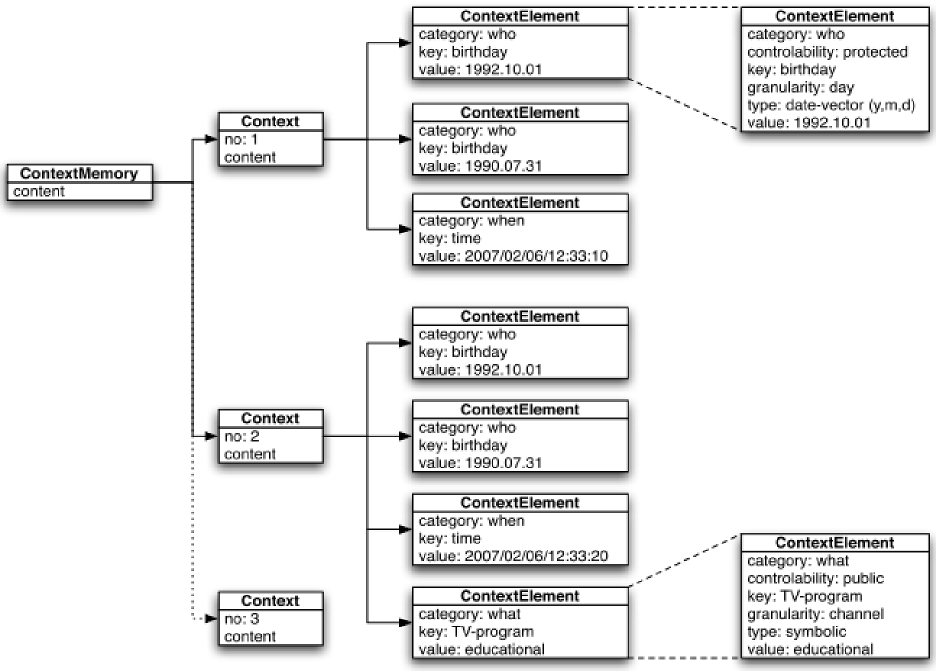

Figure 21. User- centric context model [61]

As it is illustrated in figure 21, the fundamental components which form context are **ContextElement**, **Context** and **ContextMemory**. **ContextElement** is composed of six attributes (category, controllability, key, granularity, type and value) of which the content of category could be sorted accordingly into one component of 5W1H. Context object preserves all context information which can be retrieved by suitable category and key. All useful context information which contains not only the current data but also the historical context are saved in **ContextMemory**. Users or applications can access to the **ContextMemory** through different query mechanisms.

[60] outlines the idea of user-centric context modelling clearly, here it is not going to conduct an in-depth research. The method can express context in a very organized way and it utilizes the concept of ontology which bridges the gap between context modelling and reasoning. However it is just a trade-off between complexity of expression and ease of use. And similar as other modelling techniques, it is lack of standardization.

4.3.9. Ontology-based context modelling

By tracing the origin of ontology it can be concluded that the term ontology arises from the subject of philosophy. It studies the nature of being, existence and categories of being and their relations. And further it is employed in the semantic field. Therefore, borrowed from semantic technology ontology is developed to model context. Ontology is a concept to represent a description of the attributes and relationships. Ontology-based context modelling is regarded as the most promising method to model context. This statement is widely accepted by academia and industry. So in this part, a detailed description for the ontology-based context modelling is presented in order to prove the statement.

The definition made by Studer in [61] is as follows: "an ontology is a formal, explicit specification of a shared conceptualization. A conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. For example, in medical domains, the concepts are diseases and symptoms, the relations between them are causal and a constraint is that a disease cannot cause itself. Formal refers to the fact that the ontology should be machine readable, which excludes natural language. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group."

Another explanation defined by Noy in [62] is "a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (**slots** (sometimes called roles or properties)), and restrictions on slots (**facets** (sometimes called role restrictions)). An ontology together with a set of individual instances of classes constitutes a knowledge base." This definition is made from the perspective of interior composition of ontology. Via the understanding of this definition the information about what makes up of ontology indeed can be got.

Definition for ontology in [63] is "ontologies are agreements about shared conceptualizations. Shared conceptualization includes conceptual frameworks for modelling domain knowledge; content-specific protocols for communication among inter-operating agents; and agreements about the representation of particular domain theories. In the knowledge sharing context, ontologies are specified in the form of definitions of representational vocabulary. A very simple case would be a type hierarchy, specifying classes and their subsumption relationships. Relational database schemata also serve as ontologies by specifying the relations that can exist in some shared database and the integrity constraints that must hold for them. "

From the typical definitions, it can be concluded that ontology can address the conceptual confusion among people and systems because it shares the common understanding. Besides, interoperability between different systems, devices, applications and services is another benefit enabled by ontology. In addition, it allows the reuse of domain knowledge.

Ontology models consist of Classes, Properties, Individuals and Restrictions. More specifically, they are:

- **Classes.** They refer to as Concepts which are the focus of the ontology. They define the important terms existing in the domain. The names of Classes should be straightforward to be understood in a literal sense.
- **Properties.** Sometimes they are called Roles, Relationships or Slots. They specify the relationships and attributes of classes and Individuals.
- **Individuals.** They are instances of Classes.
- **Restrictions.** They stand for the requirements for Properties.

In general, ontology can be applicable to different areas of use, more specifically, those are:

- **Domain ontology**: it facilitates the construction of knowledge for a particular domain (e.g. electronic, medical, mechanic).
- **Generic or core ontology**: the field it models is broader which usually covers several domains.
- **Application ontology**: normally, this kind of ontology works for a specific application or scenario.
- **Representational ontology**: it does not commit to any particular domain.

There are a number of semantic web languages which support the development of ontology such as **RDF** (**Resources Description Framework**), **RDFS** (**Resources Description Framework Schema**), **OWL** (**Web Ontology Language**) etc. Among all, **OWL** is the most powerful and expressive language for creating ontology. In fact, **OWL** is the extension of **RDF** with additional vocabulary and stronger syntax enabling properties, characteristics of properties (like **Symmetry**) and cardinality (like **Exactly One**). Besides, it contains a rich group of operators, e.g. intersection, union and negation.

In particular, **OWL** has three sublanguages designed for meeting the demands from different domains or applications which are listed as follows:

- **OWL-Lite**: It is the least powerful language of all three languages and as a result it is the easiest language to be used. Only basic classification taxonomy and simple constraints are allowed. It is envisioned that **OWL-Lite** is capable of migrating thesauri with other taxonomies.
- **OWL-DL**: **DL** is the abbreviation for **Description Logic**. **OWL-DL** is based on **Description Logic** so it is available for automated reasoning. It is more expressive than **OWL-Lite**. All **OWL** language constructs are included in **OWL-DL**, however only under specific restrictions they can be employed. Inconsistency in an ontology written in **OWL-DL** can be checked autonomously.

- **OWL-Full**: **OWL-Full** is regarded as the most expressive language. It allows the most syntactic freedom of **RDF** while it does not need computational guarantees. However no reasoning can meet all the features of **OWL-full** ontologies.

OWL-DL is going to be adopted to construct the ontological model later. **DL** is a group of logic-based knowledge representation formalisms including concepts, roles, individuals and operators.

Typical operators in **OWL-DL** are:

$\cap$ (Intersection), $\cup$ (Union), $\neg$ (Compliment), $\equiv$ (Equivalent), $\forall$ (All-Values-From), $\exists$ (Some-Values-From) etc.

However, ontology can be developed and formalized in **OWL-DL** in a number of software tools, such as Stanford Protégé, WebProtégé (Online version of Protégé), IBM Integrated Ontology Development Toolkit, NeOn Toolkit, SWOOP, TopBraid, OntoFly etc. These development tools enable users to design their ontologies more easily and faster.

So far, enough knowledge to prove the importance and advantages of ontology to model the context are provided. Next, more information related to ontology especially its characteristics are presented.

While developing an ontology, several key points need to be kept in mind in order to create a more proper and accurate ontology. Next, the specific requirements proposed in [64] are listed.

- **Simplicity.** The design of ontology should be simplified in expression, nomenclature of classes or in relationships to fulfill the task of the application.
- **Flexibility and extensibility**. Flexibility and extensibility which refer to scalability are important parameters needing to be taken into account while designing an ontology which means the desired ontology should be capable of supporting the addition of extra elements, classes or relationships.
- **Genericity**. The ontology should be open to support various kinds of context instead of being restricted to a single context type.

- **Expressiveness**. The description for context states should be as much expressive as possible. People can get the meaning simply by reading the ontology.

4.3.10. Comparisons of the existing context modelling techniques

After introducing the existing context modelling techniques, it is valuable to make a comparison for them. Table 7 illustrates a brief analysis of the aforementioned context modelling techniques.

This table shows the features of the actual of context modelling techniques. As depicted from the table, each context modelling technique has its advantages and disadvantages. After evaluating all of them, the conclusion is that ontology-based context modelling method could be the most promising technique to model context to achieve context awareness in IoT applications. Some publications defend the opinion that the best way to model context is to create a **novel technique** to integrate the existing context modelling techniques. This should be a solution needing more research in the future because the integration of different techniques could enhance the benefits and mitigate their respective shortcomings. However, in this thesis, the importance and benefits of ontology-based modelling are emphasized because the idea that the ontology-based modelling technique is the most promising technique so far is accepted.

Context Modelling Techniques	Pros	Cons
Key-value	Simple; Ease of use; Flexible	Lack of standards; Useless when big in size; Cannot represent relationships; Difficult to retrieve information; Lack of validation tools; Lack of scalability
Markup	Structured; Some validation tools are available; Flexible.	Lack of standards; Problems in capturing relationships, timeless, dependencies, inconsistency checking, reasoning and uncertainty.
Graphical	Rich expressiveness; Relationships are allowed; Validation is possible through constraints, Different standards and implementations are available.	Interoperability is unsolved; Configuration must be required; More standards should be established.
Object-oriented	Relationships are allowed; Some development tools are available; Can be fused by using programming languages.	Lack of standards; Lack of validation; Hard to retrieve information; Reasoning is not supported.
Logic-based	Rich expressiveness; Support reasoning; Consistency check; Simplicity; Processing tools are available.	Lack of standards; Lack of validation.
Multidisciplinary	Model context from multiple disciplines; The division of context is concrete.	Quite complex; Still at the first stage; Interoperability is unsolved; The implementation is rare.
Domain-focused	Expressive; Flexible; Structured.	Lack of standards; Lack of validation.
User-centric	Express context in a organized way; Scalability; Allow reasoning.	Lack of standards; Hard to use; Lack of validation.
Ontology-based	Support reasoning; Rich expressiveness; Relationships are allowed; Strong validation; Processing tools available; Mature standards; Interoperability.	Representation can be complicated; It will be complex to retrieve context information.

Table 7. Comparison for the existing context modelling techniques

4.4. Context reasoning techniques

Reasoning is also called inference. The demand for context reasoning derives from the features of context data which are imperfection and uncertainty [65]. More specifically, the imperfection can be categorized into four types: unknown, ambiguous, imprecise and erroneous. Because of the features of context data, context reasoning is worth being performed. In nature, the task of context reasoning is to deduce high level context from the raw context associated with some basic functionalities such as removing outliers, filling the missing value, checking context inconsistency, validating the context value and doing some simple calculation for context value. For example, at some point a camera detects a person's presence in the kitchen with the exact location L1 and his activity is cooking. After five seconds, two sensing devices are aware of the updated location for the person. One is the sensor embedded in the table located in the dining room that measures the change of pressure on the table. Integrated with other sensing devices like camera or RFID, the context shows the person is in the dining room maybe is placing the dishes (location: L2). The other sensing device is the sensor embedded in the bed and it detects the change of pressure of the bed which indicates the person is sleeping in the bedroom (location: L3). Obviously, these two location values contradict each other because at the same exact time a person cannot exist at two different places. And by making use of additional context information like the maximum speed of people, the distance between the kitchen and dining room, the distance between the kitchen and bedroom the reasoning process can solve the inconsistency and select the correct location. The example can be seen in figure 22.

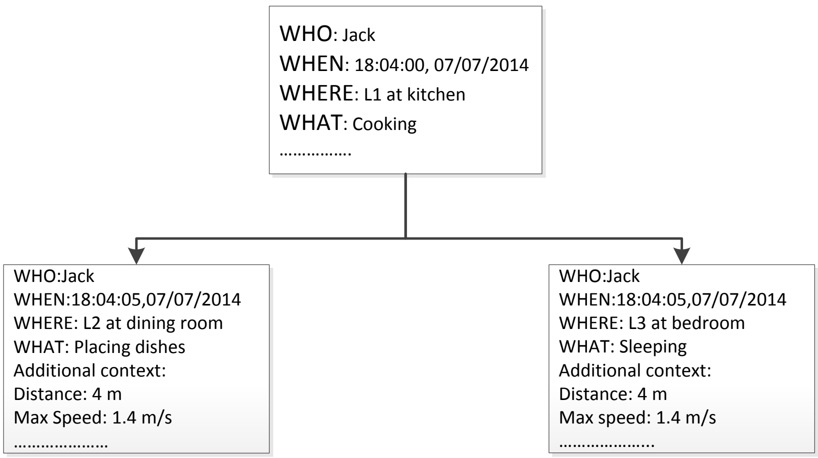

Figure 22. An example for checking inconsistency

Context reasoning is the further process of raw context. More useful context or predictions can be generated in this phase. After reviewing the literature, a brief introduction for the popular reasoning techniques is provided in the following. The whole taxonomy of reasoning techniques which are already proved to be effective methods for inferring information can be seen in figure 23.

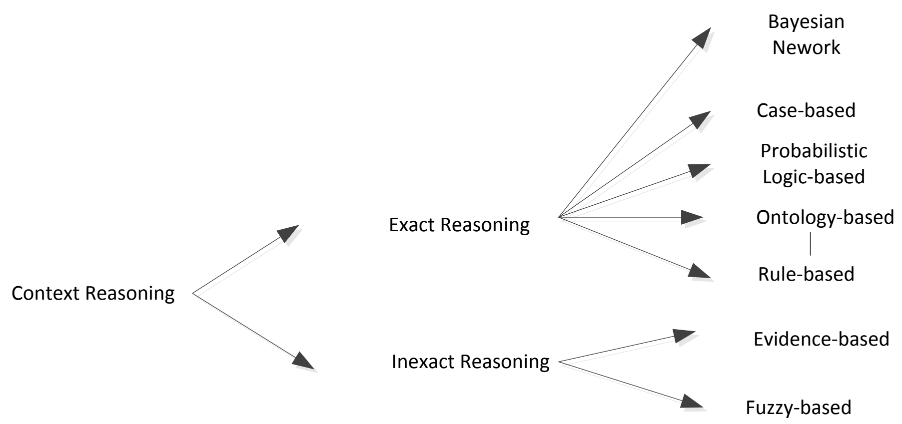

Figure 23. Taxonomy of context reasoning techniques [23]

Generally reasoning techniques are categorized into two types, namely, they are exact context reasoning and inexact context reasoning, respectively.

4.4.1. Exact context reasoning

Bayesian network [66], case-based [67], probabilistic logic-based [68], ontology-based [69] and rule-based [70] reasoning techniques are members of the exact context reasoning approach.

- **Bayesian network**. The core of this technique which belongs to supervised learning is based on probabilistic reasoning concepts. People bring in this kind of method from statistical domain. Entities and relationships among them are represented by directed acyclic graphs and probabilities. Two drawbacks limit its usage, one is huge demands for exhaustive and exclusive hypotheses and the other one is exponential computation overhead.

- **Case-based**. Literally, it deduces high context information based on the previous similar cases. It remains a challenge to calculate the accurate similarity between the current case and previous cases.

- **Probabilistic Logic-based**. A probability is assigned to make logic assertions. It allows sensor data fusion from two different resources. When conflicts occur, it can be helpful to make decisions. This kind of reasoning technique can only be applied in a scenario with the premise of probability already known.

- **Ontology-based**. Semantics are incorporated with reasoning procedure. Based on description logic, it gains supports mainly from two semantic web languages: RDF (S) and OWL (2). Ontology-based reasoning can be executed for achieving more useful context information based on predefined classes and relationships. However, it is lack of capabilities of finding missing value or ambiguous information with compensation of expressiveness and generating new context information.
- **Rule-based**. System infers high level information on the basis of rules pre-defined by users or designers. The disadvantage is that it cannot generate new rules adaptively according to varying situation.

4.4.2. Inexact context reasoning

- **Evidence-based**. [71] proposes an evidence-based reasoning method known as Dempster-Shafer theory. It releases two assumptions held by Bayesian network and allows inferring context that are not based on the rules or semantics while are limited by intensive computation.
- **Fuzzy-based**. The fuzzy set theory is explored in this reasoning technique which degrees of membership can be represented by confidence values. In traditional logic, precise numeric value ranging from 0 to 1 is used to represent truth values. More natural expressions are allowed in the fuzzy logic reasoning to extend the ability to handle uncertainty. Nevertheless, the drawback is that all the uncertainties need to be predefined manually.

4.4.3. Conclusion for context reasoning techniques

To conclude, inexact reasoning strategies are less used in real applications due to the fact that they cannot infer the accurate implicit context from raw context. In general, the selection of context reasoning technique is subject to two factors: the performance of the reasoning technique and the requirements from the modelling technique which is being used in the same scenario. The previous descriptions for the existing reasoning techniques show that the ontology-based reasoning is a powerful solution to deduce high level context based on the available context because of its predomination of knowledge sharing, logic inference and knowledge reuse. In fact, different reasoning techniques have restrictions for context modelling techniques because of their individual features. At present, random combinations of any context modelling and reasoning techniques have not been realized. To design a specific application, the selection for the modelling and reasoning technique should be made carefully taking as much criteria and requirements as possible into consideration. In the thesis, ontological context modelling and reasoning techniques are adopted as the major and crucial techniques to construct and infer context.

5 Context awareness frameworks for IoT

In this chapter, the main interest focuses on the context awareness frameworks for IoT. At the beginning, a brief investigation on the well-known context awareness frameworks is conducted which are prototypes derived from those accomplished projects related to context aware IoT applications. Afterwards a proposal for context awareness framework is given along with detailed explanations of important modules.

5.1. Research efforts in context awareness frameworks

A number of context awareness frameworks intended to be applied in IoT have been developed by different researchers. The efforts merged in this area are noteworthy. And some of them have already been evaluated and proved to be mature to some extent while the others need more research efforts put on them. In this subsection, several popular context awareness frameworks are introduced.

5.1.1. Cobra (Context Broker Architecture)

A new smart meeting room was developed in [72] which provides a number of intelligent services. This smart meeting room was built in compliance with a broker centric agent architecture which is named **Cobra** (**Context Broker Architecture**). The central component of **Cobra** is **Broker**. A context broker takes responsibilities as follows:

- A centralized model of context is available for all devices, services and agents to share.
- Require contextual information from sources which are not available from resource-limited devices.
- Context reasoning is realized.
- Context inconsistency can be checked.
- Privacy is taken into consideration by enforcing policies predefined by users to manage context sharing and reuse.

The whole picture of system diagram of **Cobra** can be depicted as: a context broker is the essential component of the whole system and it obtains context from heterogeneous sources and makes further process on the context. More specifically, the inner structure of a context broker can be seen in figure 24.

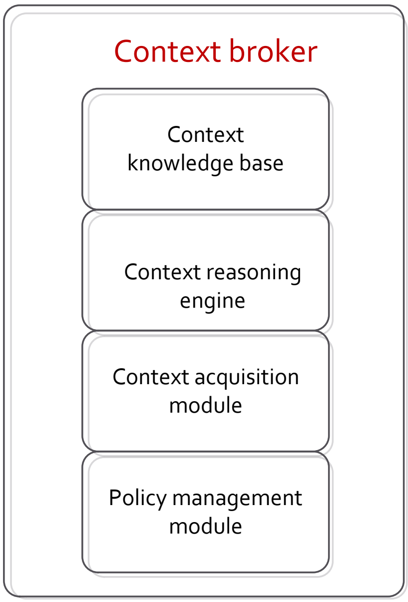

Figure 24. The structure of a context broker

As illustrated in figure 24, four major components constitute a context broker, namely, they are **Context knowledge base**, **Context reasoning engine**, **Context acquisition module** and **Policy management module**. Furthermore, the functionalities of these modules are identified in the following.

- ***Policy management module***. Policies defined in this module are taken into account before a context broker shares context information with other agents. Policy determines the rights endowed to each agent. So different agents can only be allowed to consult the context information which is assigned to them.
- ***Context acquisition module***. The goal of this module is to obtain context information from various resources.
- ***Context reasoning engine***. Ontology inference on the basis of OWL's semantic and domain-heuristic rules is employed in this module aiming at extracting more useful context information.
- ***Context knowledge base***. All context information are formulated in RDF triples (subject, predicate and object) and stored in a database.

The major difference between **Cobra** and other similar frameworks is that ontology expressed in OWL is adopted to support context modelling, sharing and reasoning. Besides, **Cobra** shows its pioneering efforts in protecting privacy. All the contributions made by **Cobra** can inspire other researchers to improve the ongoing work on context awareness frameworks. However, there are still some challenges remaining to be addressed. The future work can be focused on three aspects:

- Solving the scalability of knowledge sharing especially in a dynamic environment.
- The performance of context reasoning should be improved as the increase of sensing context volume.
- The way to specify the policy should be figured out.

5.1.2. Context Toolkit

Context Toolkit [73] is another framework with the intention of supporting the rapid prototyping of context awareness applications. Java was employed to develop **Context Toolkit**, however it still allows the creation and interoperability of different widgets written in any language. The figure 25 extracted from [73] shows the general structure of **Context Toolkit**. Several abstractions are created to form the **Context Toolkit**.

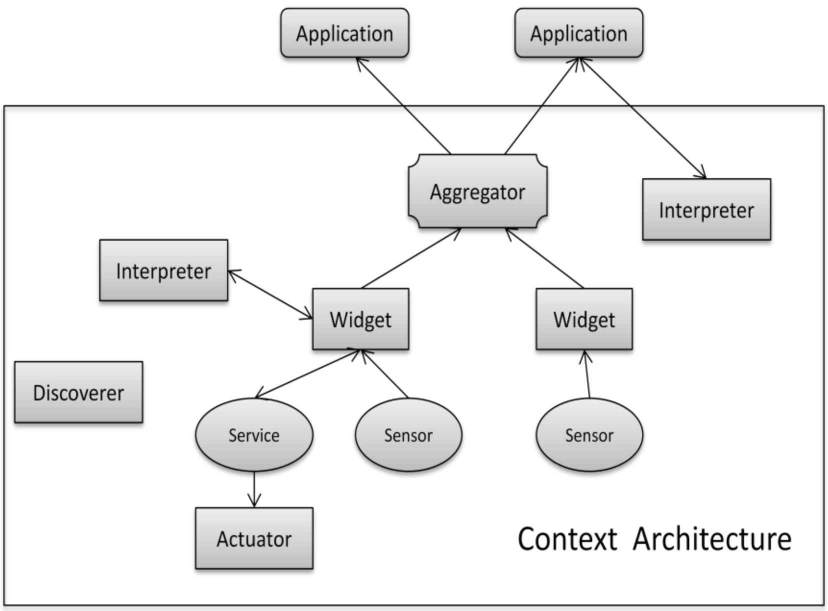

Figure 25. An example configuration of Context Toolkit components

- **Context Widgets**. Roughly speaking, a context widget is a software component that makes it possible for applications to access to context information. Users are capable of interacting with the applications.
- **Interpreters**. This component is designed to express context in a higher abstract level. It is very significant for the whole framework because most applications are eager to get high-level context information instead of raw information directly detected from sending devices.
- **Aggregators**. Just as the name implies, aggregators work for integrating relevant context information for simplifying the vast number of context information. Aggregators provide a lot of aids for architecture to deliver aggregated pieces of context to the specific application accordingly.
- **Services**. Context services are responsible for executing actions after making use of the context information. They can be either synchronous or asynchronous.
- **Discovers**. As the last component of this framework, the module of discovers is in charge of registering all the entities which are existing in the framework. They get the awareness of deployment of components in real time. For example, they know what widgets are developed and available in the current situation. Besides, discovers can help find a similar component to substitute for the unused one.

The concept of context ownership is introduced to confine user to access the context which they are not allowed with in **Context Toolkit.** However, a few context aware applications have been exploited based on this framework because **Context Toolkit** cannot meet all the requirements of building a context awareness framework [73]. However, due to the fact that this context awareness framework is started to be constructed, the issue how to develop more context aware applications upon this framework should be addressed in the very near future.

5.1.3. Gaia

Gaia [74] is a distributed middleware infrastructure which is designed to realize **Active Spaces** [75]. The **Gaia** framework is extracted from a project with the same name. In this project, all physical spaces are conceived as **active spaces** with the awareness of all contained resources and their status. In **Gaia**, context information is expressed as ontology and rules are specified in first-order logic and Boolean algebra, as a result, context can be easily understood and inferred. It enables users with services to query, access and make use of existing context. The objective behind this is to support user-centric, resource-aware, context-sensitive and mobile applications. Six key components constitute **Gaia**, namely, they are: context provider, context consumer, context synthesiser, context provider lookup service, context history service and ontology server. The name of each component is quite concise and straightforward which their functionalities can be understood more or less just in a literal sense. In general, the major contribution of **Gaia** is the interaction among different services. It is also worth noting that the deficiency of **Gaia** is that security service should be merged into the whole architecture. Last but not least, more context aware applications should be developed in order to validate the performance of **Gaia**.

5.1.4. SIM (Sensor Information Management)

SIM (**Sensor Information Management**) was proposed in [76] aiming at addressing tracking location. This framework facilitates interference and conflicts within the sensed information. A priority-based mechanism is employed to solve the conflicts in SIM, each node is distributed with an associated priority. Generally, **SIM** is composed of two basic parts: sensor platform and intelligent agent platform. More specifically, in the sensor platform, a variety of context information are gathered and aggregated. The sensing devices like sensors are connected to each other in a star network topology. All context information which come under integral management by **SIM** can be stored in a database called the information base. Besides, the agent-based platform conforms to the standard specifications provided in **FIPA** (Foundation for Intelligent Physical Agents). An algorithm to improve position accuracy is also put forward in this article. In fact, **SIM** is not ideal since it has some remaining issues needing to be addressed. First of all, the information conflict within identical sensed information and services is worth being paid attentions. What's more, security and privacy is an obstacle needing to be conquered in the future.

5.1.5. Aura

Aura [77] emphasizes figuring out the method to overcome the bottleneck of computing, namely, the limited resource of human attention since human attention is a scarce resource. In order to accomplish this goal, **Aura** covers every system level: from the hardware, through the operating system, to applications and end users. Two shining features of **Aura** are as follows: proactivity and self-tunning. More specifically, **Aura** enables a user to maintain continuty from a higher level. Besides, layers are able to adapt their behavior such as adjusting their performance and resource usage characteristics according to the observing demand. **Aura** considerably makes progresses in supporting user mobility and preventing users from being affected by variation in resource availability. In general, **Aura** consists several main modules as depicted in figure 26: Intelligent networking (Network weather monitoring, network proactivity), Linux kernel, Coda (Nomadic file access), Odyssey (Resource monitoring, adaptation), Spectra (Remote execution), Other Aura runtime support and upper application layer.

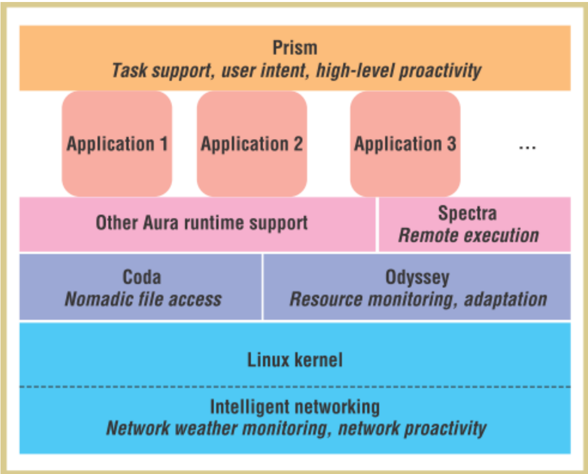

Figure 26. The architecture of Aura from [77]

Odyssey embodies context awareness in Aura. Four major components make up this module, they are context observer, task manager, environment manager and context suppliers, respectively. XML-based scheme is adopted as the context modelling technique.

5.1.6. MoCA (Mobile Collaboration Architecture)

MoCA (**Mobile Collaboration Architecture**) [78] is a flexible and adaptable context aware framework extracted from the project with the same name. Context meta-information is adopted to improve the performance of context awareness in **MoCA** despite of the environment's heterogeneity and evolution. **MoCA** employs meta-information to make decisions in middleware level. The context awareness infrastructure is roughly made up of three components, namely, they are context providers, context consumers and context information services. Context provider is the module which copes with context like probes raw context or aggregated basic context. Then context consumer is the entity which is interested in the context and able to consume the context. Context provider and context consumer are linked by context information service which is responsible to receive, store and disseminate context to consumers. In **MoCA**, all types of context are modeled in XML because of its strong expressiveness. The future work of **MoCA** is pointed out that the framework should be extended flexibly to accommodate quality-of-context parameters.

5.1.7. HCoM (Hybrid Context Management)

HCoM (**Hybrid Context Management**) [79] is a collaborative platform that uses semantic ontology approach to manage context semantics and relational approach to manage context data. With the combination of semantic ontology and relational schema, the performance of this model is dramatically improved because it overcomes the limits with respect to only using one of them. In particular, standard database principles cannot totally manage context data. Besides, the efficiency and query processing of semantic ontology are undesired. All these reasons are the motivations to drive the proposal of this hybrid approach. **HCoM** is designed as a layered architecture within several functional modules. From the lowest to the highest, those layers are acquisition layer, preprocessing layer, data modelling and storage layer, management modelling layer and utilization layer. Generally speaking, it is easy to get the understanding that the major novelty of **HCoM** is to combine semantic ontology and relational schema. Besides, a heuristic based selective algorithm enables the scalability of this framework by loading the context data into its reasoner with respect to relevance. However, the research on **HCoM** is not mature so far. Further work like validation, evaluation etc. needs to be conducted.

Table 8 examines the features of the discussed frameworks. This list is not an exhaustive review of existing frameworks, nevertheless, they represent the current development status of context awareness frameworks. It is worth noting that all of the existing frameworks have defects needing to be addressed. Reminding the description and analyzing the characteristics of aforementioned context awareness frameworks, it can be summarized that the common deficiencies of them are as follows:

- **Security** and **privacy**. Security and privacy are significant issues and need to be taken into consideration in any system. Especially in IoT scenarios, contexts related to private information are employed. Hence, security and privacy should be guaranteed in different levels in IoT. Currently the majority of existing context awareness frameworks cannot cope with the requirements from security and privacy.
- **History** and **storage**. Context history is meaningful in context awareness applications. For example, based on the context history, more useful context can be deduced. How to store context is a concern necessary to be figured out. Besides, because of the scale of IoT, it is not feasible for context to be stored for the long term. Some frameworks have taken this issue into account by adding storage devices like **Cobra.** On the contrary, history and storage are still missing in some other frameworks such as **Aura**, **MoCA** etc.

Framework	Architectural style	Modelling	Reasoning	Scalability/ Adaptability	Interoperability	Security and Privacy
Cobra	Centralized	Ontology (OWL)	Rule, ontology	×	×	√ (User can define policies by Rei policy language)
Context Toolkit	Widget/Component based	Key-value	Yes, but not specified	×	√	√ (The concept of context ownership)
Gaia	Distributed	Ontology	Fuzzy logic, probabilistic	√	√	√
SIM	Distributed	Key-value	Rule	×	×	√
Aura	Modular	Markup (XML)	Rule	√	×	×
MoCA	Centralized	Ontology	Ontology	√	√	√
HCoM	Layered	Ontology	Ontology, rule	√	×	×

Table 8. Summary of the discussed frameworks

5.2. Proposed Context Awareness Framework

Aiming at solving the aforementioned issues and identifying the major functionalities in context awareness, in the following, a context awareness reference framework for IoT as shown in figure 27 is proposed. The framework is logical which means each component's actual location or distribution is not described. However, the well-defined responsibilities and functionalities of each component are established in the following.

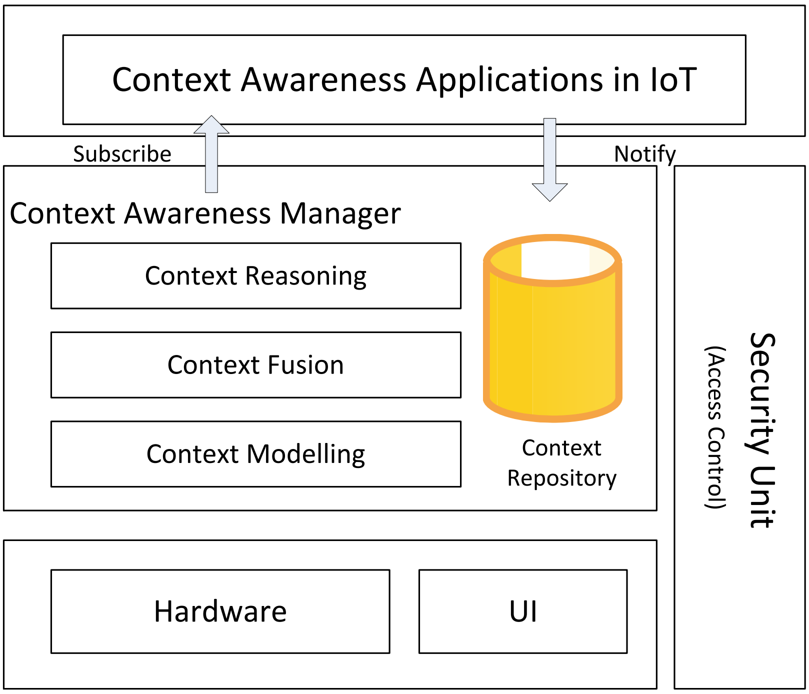

Figure 27. Proposed Context Awareness Framework

It is revealed from the full picture of the whole architecture that seven crucial modules constitute this framework and play significant roles in facilitating context awareness in IoT. Next, introduction for each module associated with functionalities and schema are presented separately in detail.

- **Hardware**. The framework is built on the top of a hardware and a **UI** (**User Interface**). The hardware is the collection of a variety of devices with the abilities of sensing, computing and communication etc. Sensors, actuators and RFID are typical devices widely used in the field of IoT. Indeed, different kind of hardware has different functionalities and responsibilities. In essence, hardware is the main source for capturing context data and therefore is the basic component in the framework.
- **UI**. **UI** is the abbreviation of **User Interface**. Especially **HMI** (**Human-Machine Interface**) or **HCI** (**Human-Computer Interface**) are highlighted in this module. The way that users act with computers or other related devices is easier, more interesting and more friendly than the conventional interfaces. Users are capable to check context information they are interested in and in the meanwhile can input their private context information such as user's profile via **UI**.

- **Context modelling**. Context modelling refers to mapping the context. It is the stage to represent the object "context". More specifically, context information obtained from the heterogeneous sources is structured in a unified format in order to enable a system to understand the meaning of context. In previous sections, the popular context modelling techniques have already been investigated. As a result, ontology is emphasized as the most promising technique to model context. So in this module, ontology is adopted to represent context. Since a thorough description for ontology-based context modelling is presented in the previous chapter, here the relevant information about ontology-based context modelling techniques won't be repeated.
- **Context fusion**. In IoT, more specifically, in some context awareness scenarios, a number of nodes are used to cover the same region of interest, each cooperating with a partial view of the scene. Redundancy is a challenge in IoT calling for more efforts to focus on itself. The process of context fusion is proposed to deal with this issue. In this module, overlapping measurements can be fused to obtain more accurate context information. Different sensors can monitor different properties of the same environment. In this way, complementarity is achieved which can represent all the changes in the environment from different aspects. Context fusion is capable of combining complementary context information which is beneficial for inferring more useful context. How the contextual information gathered by sensing devices can be processed to increase its relevance is concerned with by context fusion module. Context fusion can be classified into three categories according to [80]: based on relationships among input context; based on abstraction level of manipulated context during fusion process and based on the abstraction level of input and output of a fusion process. [81] investigates a few of mechanisms and algorithms which can be employed in context fusion.
- **Context reasoning**. Context reasoning refers to deducing high level context from the results from the previous stages The module of context modelling and context fusion supply sufficient raw context information to context reasoning which make the

reasoning development smoother. Obeying predefined rules, more useful context information can be generated on the basis of the available low-level context. The high-level context information is more informative to the context awareness application compared with the basic context directly obtained from the environment. An ontology-based context reasoning technique is adopted in this module due to its notable benefits. And ontology is the key technique being accepted in the remainder of this thesis.

- **Context repository**. A context repository is similar to a knowledge base proposed in other literature which is responsible for storing all context information related to the specific domain. Due to the features of ontology information saved in the repository can be reused. Sometimes the historical context data can be used to establish trends and predict future context values. Indeed, the main concern about the maintenance of context historical data is memory. The **Context Toolkit** and **Cobra** mentioned save all the context data persistently in a database. An additional advantage of using a database is that it can make use of the SQL which allows reading and manipulating operations at a high abstraction level. Besides, it is possible for a context repository to store more than one ontology since ontology allows inheritance. When dealing with a complex application, it is brilliant to create a small ontology for a specific topic, use inheritance from mature ontology and get a composition instead of tackling directly the whole design of complex ontology.
- **Security unit**. This module is significant to guarantee users' security and privacy, however, many existing context awareness frameworks do not take it into consideration, and a few frameworks just provide basic security mechanisms. In order to provide more customized services in context awareness application, more sensitive context information related to a user's privacy is collected and used. The concern hidden behind the usage is that once those information is leaked out, it could be dangerous or a threat for the involved persons. Especially in healthcare environment, involved in it, amounts of private information like medical information, physical information and personal preference are used in order to help doctors and caregivers

make better decisions. So the associated security risks should be minimized. Measures should be taken to protect security and privacy, such as assigning context data to their respective owners. Users' access to context data should be carefully controlled. Only authorized users can be granted to access to context data meeting with auxiliary requirements such as the time constraints, the location constraints etc.

- **Context dissemination**. Context is meaningful for context-awareness applications to adapt their behaviors according to the ever changing environment. Context dissemination is the process that distributes the context information from the previous modules to the context-awareness applications. The mechanism adopted by this module is: applications subscribe to the specific events of interest, once any change in the interested area is detected, the occurrence of the events of interest is notified to the applications.

6 An ontological context awareness model for the home healthcare scenario

As discussed earlier, IoT enables the booming developments in many domains such as medical area, military area, transportation area etc. Nevertheless, the major interest of this thesis will be limited to home healthcare. Healthcare is not a new topic, on the contrary, judging from current literature it is easy to note that healthcare is quite a popular field drawing a lot of attention from industry and academia. In fact, healthcare is a general concept which refers to delivering the "right treatment to the right patient at the right dose and at the right time" [82]. The traditional model of healthcare mainly focuses on improving medical services in hospitals due to the increasing efforts put in the field of hospital healthcare. However, the ongoing demographic changes in the worldwide population are considerably stressing the current healthcare system with the increasing number of elderly and chronically ill patients. Apart from the huge burden brought by the changes in population, the fast growing demand from elderly and chronic patients which is desiring to live as independently as possible is another factor that paves the way to alter and overload the pattern of existing healthcare services. Independent lifestyles come with more risks and challenges. As a consequence, home healthcare is a potential solution to deal with the rising risks. In [83], authors make a detailed research on the development of home healthcare and they conclude that the need for it is being driven by several factors. More specifically, they are demographic trends, the needs of patient in-home healthcare, the unbalance between patients' needs and medical resources and changes in healthcare towards more cost-effective approaches. A typical and traditional home healthcare model is hiring a nurse who can provide various levels of care. This kind of home healthcare is somewhat called one-to-one healthcare which requires a fixed medical professional to monitor the patient's activities and conditions in the home. In general, two major disadvantages limit the usage of this model. The first problem is hiring a nurse is quite expensive and not every family can afford the high expense. Furthermore, current medical resources cannot live up to the increasing demands of patients. The second problem is that the privacy of patients cannot be guaranteed because some patients should be monitored all the day. All the aforementioned drawbacks give motivations to improve the home healthcare services. The desired goal is to ensure that elderly people and chronic patients can live safely and independently in their own home as long as possible. In this context, intelligent home healthcare system for many uses is proposed in many publications as the promising solution to meet the needs identified above.

In the following, the context awareness framework proposed in the previous chapter will be applied into the home healthcare area. A specific home healthcare scenario will be described along with listing its main functions. Besides, a formal approach to create an ontology will be presented. By making use of it, an ontology aiming at representing the specific home healthcare scenario will be generated and explained in detail. In general, the major contribution in this chapter is to create an ontology for a specific home healthcare scenario by adopting the proposed context awareness framework and formal method to develop an ontology.

6.1. Formal approach to design an ontology

It is widely accepted that there is no "correct" way or methodology for developing ontologies [84]. In this context, so far, various ontology development methodologies have been presented in many publications such as On-To-Knowledge and METHONTOLOGY etc. However, all those approaches mentioned previously only work under specific conditions, they are lack of generality and universality. The first methodological outlines were proposed on 1995. Pinto and Martins [85] put forward a general ontology design process with five stages including: specification, conceptualization, formalization, implementation and maintenance. Another knowledge-engineering methodology is proposed in [86] which refers to a simple guide to develop ontologies. Seven steps form the whole process. More specifically, the detailed procedure is outlined in figure 28:

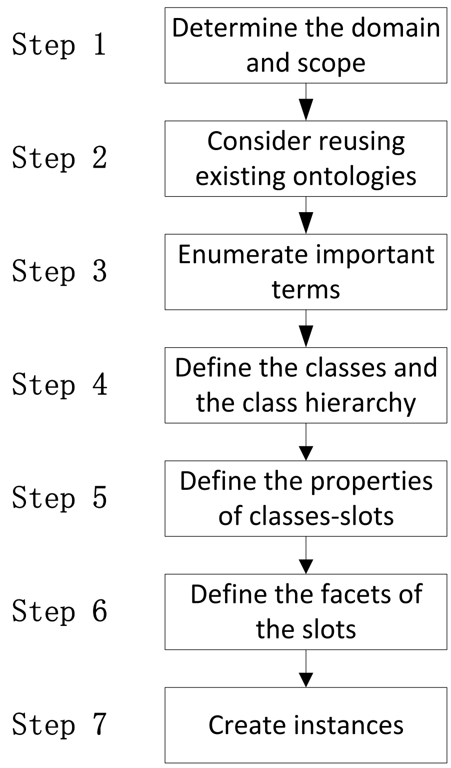

Figure 28. A simple approach to create ontologies

Apart from the aforementioned approaches, in the domain of enterprise modelling, some more ontology developing methods are reported such as Enterprise Ontology and the Toronto Virtual Enterprise.

Among the existing ontology developing approaches, the idea extracted from a dissertation named “An ontology based framework for modelling healthcare teams” will be accepted and employed in this thesis. The approach is adapted from **TOVE（Toronto Virtual Enterprise）** [85] methodology. Its main purpose is to model enterprise processes and supply chain management.

Before seeing more details about the approach adopted in the thesis, TOVE providing guidelines for developing ontologies which will be introduced first. In brief, TOVE is composed of six phases.

1. **Motivating scenarios**: the start point is to specify the main scenario which draws attention and is the potential interest. The true meaning of this stage is to define the application or specific problems in order to analyse the scope of ontology.

2. **Developing informal competency questions**: list as many informal competency questions as possible needing to be answered in this stage. Those informal competency questions can be asked based on the motivating scenario.
3. **Terminology specification**: in the third stage, main objects, concepts, attributes and relations should be formally extracted and specified (usually in first order logic).
4. **Developing formal competency questions**: the requirements of the ontology are formalised based on the formally defined terminology in the previous stage.
5. **Axiom specification**: definitions of terms and constraints are specified and guided by the formal competency questions over the ontology terminology. Actually, the axioms must be necessary and sufficient to express the competency questions and their solutions.
6. **Evaluation of ontology completeness**: the final stage acts as an evaluation stage that assesses the competency of the ontology. The competency questions from stage four will be checked. This stage represents the condition under which solutions to those questions are complete as well.

TOVE approach focuses more on the evaluation stage compared with other ontology developing approaches. In order to properly support the research scope which is home healthcare in this thesis, TOVE is selected to be the basis of the approach which will be employed later because its simplicity and generality. As talked early, TOVE will be modified from six stages to four stages in order to fit into the home healthcare area appropriately, in other word, the adopted ontology developing approach is composed of four major stages as illustrated in 29 which will be explained by outlining specific tasks, methods and outputs in detail in the following**.**

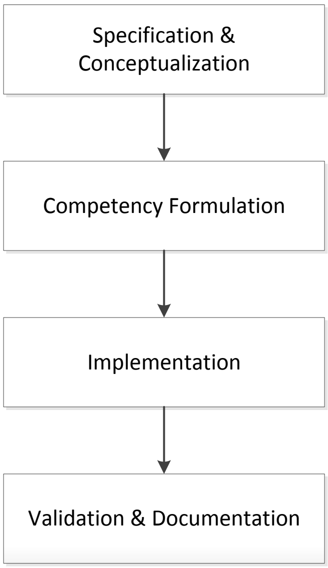

Figure 29. Employed ontology developing approach

- Stage 1-**Specification & Conceptualization**. The purpose and scope of ontology will be identified and determined as the main job of specification in this stage. Conceptualization is to define the concepts, vocabulary and relationships for ontology design. Two major tasks which are data collection and analysis are done in this stage. All the tasks in this stage are executed based on raw data so that they are textual in representation. Different data sources are able to be used according to the project context. In general, the main contribution in stage is to identify the ontology scope and declare the requirements and needs of user.
- Stage 2-**Competency Formulation**. Competency is formulated in this stage. As mentioned previously, competency questions are a list of questions sketched from a knowledge base that the ontology should be able to answer. Judging from competency questions, the ontology can include more information and refine its scope. Competency questions are specified and represented in textual level the same as the first stage. Obviously, competency questions are the only outcome in stage 2.
- Stage 3-**Implementation**. Literally, the formal model of ontology is developed in this stage while formalization is the most important part. The first step in formalization is to develop the domain ontology which is namely a formal model of concepts and categories from selective coding. The selective codes are a key for ontology design

because it acts as the starting point as they identify the scope of the ontology. Then the selective codes can be extended by developing hierarchies and relationships such as IS_A and PART_OF relationships for the ontology. In order to formalize the ontology, a set of ontology development tools can do help such as Protégé, OntoEdit and Otolingua etc. In similar, the representation languages are various including frames, OWL, RDFS etc. These tools simplify the design of ontology. Later, Protégé will be selected as the tool to develop the proposed ontology in the thesis.

- Stage 4-**Validation & Documentation**. This is a significant work because the success of creating an ontology is subject to the evaluation result. The accuracy and completeness of the ontology should be validated. For example, basic sentence error and consistency among different entities and relationships must be checked. A plug-in called Pellet embedded in Protégé can examine the consistency of ontology automatically. On the other hand, user can evaluate the ontology manually like assessing different components, including the application for which the ontology is developed for. After getting through the verification stage, documentation of introducing the ontology which can provide good feedback for getting a better understanding of the problem area or model revision or maintenance should be written down.

6.2. Overall description of IoT-based home healthcare scenario

Many efforts have been proposed for modelling home healthcare, however, most of them only focus on one or a limited set of aspects and sometimes the outcomes even present overlaps. In the following, a specific scenario is described and its main functions are demonstrated. After that, in the next section, an ontology-based model is presented for the described scenario which contains multiple services of care.

The emphasis is put on the topic of home healthcare which is selected as a test environment for showing how context awareness can be achieved in IoT applications. The home healthcare system is proposed as an effective way to provide care for patients who are living alone, which is a concern that is achieving much attention lately. With such a healthcare environment, patients would not need to have the medical services nearby. Many benefits brought by home healthcare system have been already introduced previously. To some extent, home healthcare systems may occupy the place of traditional treatment modes as they do not undergo the constraints of limited medical resources. The ultimate purpose of home healthcare systems is to make the best of medical resources. Generally, home healthcare care serves the services by identifying the situation and condition which the elderly or chronic patients are in, evaluating their medical condition and then deciding if any measure needs to be taken.

The overall architecture of the proposed home healthcare system is illustrated in figure 30.

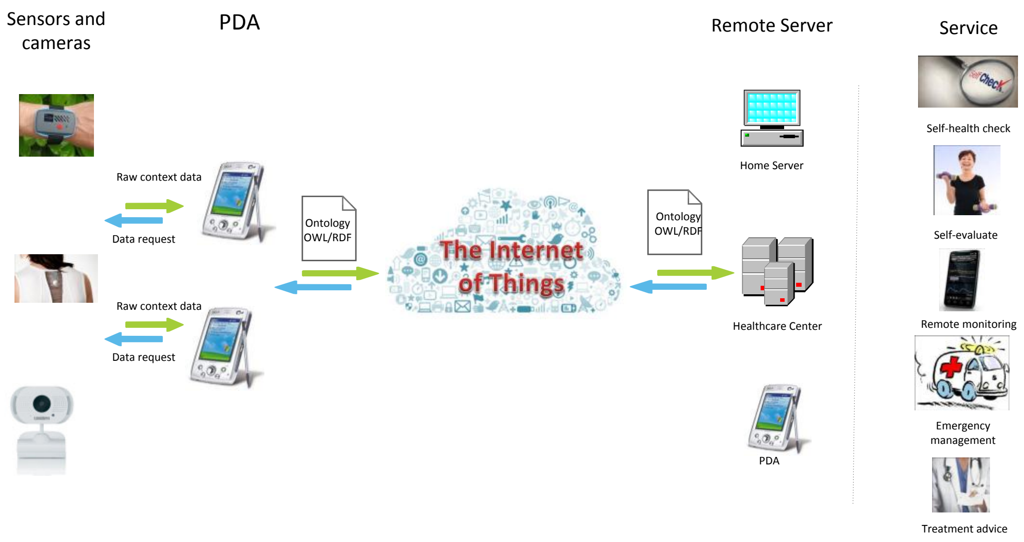

Figure 30. Overall architecture of proposed home healthcare system

The rough prototype of the proposed home healthcare system is shown in figure 30. A patient living in his home is able to monitor his health all the time and get first aids from professionals in time. Self-management for health can be guaranteed. In order to realize the ideal state, series of preparatory works need to be accomplished. Various sensors (infrared, audio, accelerometer, physiological etc.), cameras and other sensing devices should be deployed inside the house. PDAs (Personal Digital Assistant) are used for collecting raw context data, aggregating them and mapping them in a semantic manner where OWL is adopted to structure all the context data. The OWL/RDF could be stored in a ontology repository for users to make queries about the historical context information. Finally, PDAs send the OWL/RDF file representing the home healthcare ontology to remote servers. PDAs also act as an interface to import personal data like a patient's profile. Remote servers have the rights to inquire and process the sensed data. In a hospital end, medical professionals can monitor the real-time changes about the patient's health condition based on the OWL/RDF file they receive and take measures accordingly such as making prescriptions or renewing treatments. Also, the home healthcare system can provide more services with the collaboration with patients' relatives. Generally speaking, all types of elderly and chronic patients like cardiac patients, stroke patients, psychological patients etc. can get benefits from the home healthcare system. In general, four major services are offered which are self-check service, self-evaluate service, remote monitoring and emergency management.

- **Self-check service**. Basic physiological data like skin temperature, blood pressure, heart rate etc. can be detected in real time, patients can view their value on a PDA or directly on the devices. Abnormal changes can be easily detected and draw people's attention. For instance, a person who is suffering from obesity needs to check his weight every day, once he finds a increase in his weight, it is obvious that his diet or exercise plan does not work. Self-check service gives the patients the first hand information about their health condition, in addition, it provides doctors with patients' basic data. Patients can do help to monitor their own health condition rather than only relying on machine or doctors.
- **Self-evaluate service**. Patients can evaluate their performance for specific treatments and send it back to doctors as a feedback. The key role of a self-evaluate service is to help the doctors to adjust their prescriptions for drugs or exercises. Good performance proves the effectiveness of the prescription. For instance, a stroke patient follows an exercise plan which is holding a bottle and placing it to a shelf designed by his doctor. He can measure the failure times and stability of his hand by himself, at the same time data from sensors and cameras can be used to infer patient's performance as well. Cameras deployed inside the home can record the patient's performance and store it for further observation.

- **Remote monitoring**. The unbalance between medical resources and patients means that it is impossible for patients to have the supervision from their own doctors nearby. Remote monitoring is a good way to fix the gap. Doctors or other professionals are authorized to monitor all patients' activities at home and give them the necessary supervision remotely. The essence of this service is to let patients have necessary aids from doctors to keep their exercises on the normal track.
- **Emergency management**. While dealing with abnormal phenomenon like dramatic changes in the body index or a sudden faint, an emergency call can be made by the PDA to the emergency center. In fact, different measures will be taken according to the severity of the emergency. Sometimes, local alarms will arise to ask for help instead of making calls to an emergency centre because of the detection of people nearby. In general, three emergency measures are clearly designed in our system: two are already mentioned, the third is the possibility to send SMS to patients' relatives when an emergency occurs. This service ensures that the patients get first-aid treatment in time even if nobody notices the emergency.
- **Treatment advice**. This service could be an interesting improvement on the conventional healthcare mode. The home healthcare system is capable of generating the treatment advice with respect to the context from the ontology repository. The automatic proposal of treatment should take the patient's preference and medical history into consideration. These treatment advices from the system provide rich information and options for the doctors to refer when they make prescriptions.

All the content above makes descriptions clearly for the specific scenario providing sufficient knowledge base for the following ontology design. So far, all the preparatory work for developing an ontology has been provided. Next, the focus is going to move to design the ontology-based model for the described scenario and present details about the ontology.

6.3. Preliminary implementation

Chapter 4 makes a comprehensive survey on the existing methods to model context information and theoretically proves that ontology-based context modelling is the most promising approach. The major work in this section is to explore ontology-based context modelling to create a model for a specific scenario which is related to the home healthcare scenario elaborated in the previous section. As a consequence, the ontology design is dedicated to a specific application or scenario instead of domain.

Keeping in mind that the following ontology belongs to the kind of application ontology, all the design should be limited to the specific scenario by making use of the formal approach introduced in 6.1.

Section 6.1 presents a formal ontology developing approach based on TOVE along with the introduction for several existing methods. The proposed ontology developing approach lays a good foundation to create a new ontology for a specific home healthcare scenario presented in the previous section. As mentioned before, TOVE methodology is adapted to the home healthcare area and also modified its six stages to four stages including: Specification & Conceptualization, Competency Formulation, Implementation, Validation & Documentation. It is believed that the proposed ontology developing approach can be an efficient solution to develop ontologies especially suitable for home healthcare area. The following section provides more details about the ontology-based model for home healthcare scenario created in the thesis.

Following the aforementioned ontology developing approach step by step, an ontology-based model could be developed. However, it is redundant to introduce the ontology obeying the same sequence as the design process. Basically, details about the ontology-based model will be presented and organized as:

Section 6.3.1 posts the model figures directly along with descriptions for classes and relationships between them. This part of content presents the results from the first three stages of developing ontology. More specifically, it is split into two subsections: the ontology modelling part and the ontology reasoning part to make the description more clear and easier to understand.

Section 6.3.2 shows the concrete process of validation.

6.3.1. Ontology-based model

Protégé known as a free, open-source ontology editor and framework for building intelligent systems which is developed at Stanford University is adopted to design the ontology for this application. Users can edit interactively ontologies and knowledge-bases within Protégé and assess them with a graphical user interface and Java API. In this work, Protégé version 4.3 is adopted as the tool to develop the home healthcare ontology. Figure 31 shows the general interface of the Protégé software.

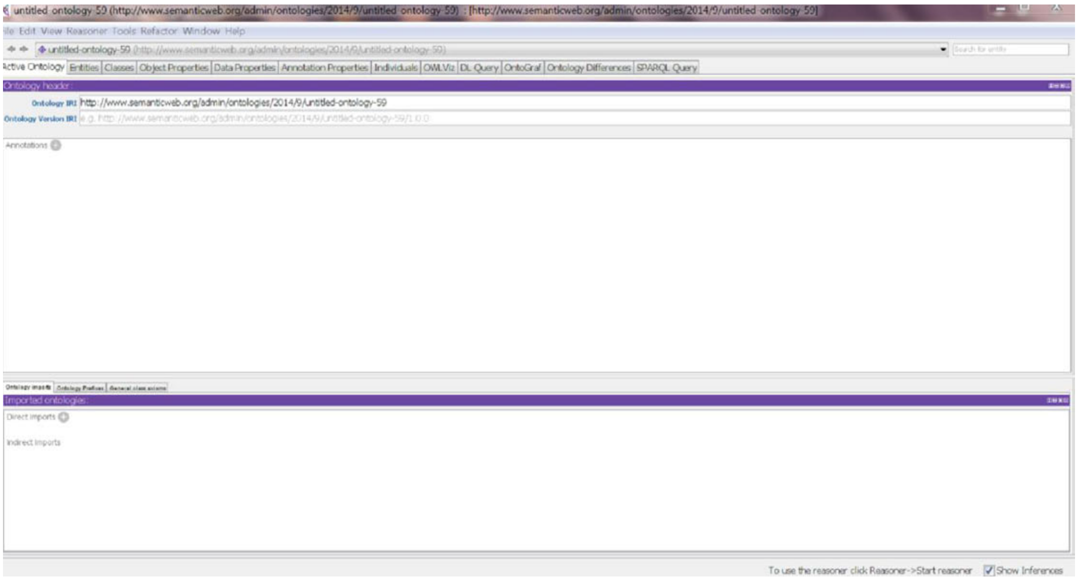

Figure 31. The interface of Protégé version 4.3

Commonly there are three tabs widely used among the several tabs in Protégé depicted from figure 31: the first one is the *Classes* Tab used to define classes, manipulate the class taxonomy and distribute restrictions over the ontology terminology. The second tab is *Object Properties* which can work for creating property hierarchies, linking different classes, specifying the characteristics and domain as well as the range of the properties. The last tab is *Data Properties* linking classes to their corresponding individual value. A prominent feature of Protégé software is that it can be extended with pluggable components to support new functionalities and services. An increasing number of plug-ins are invented to offer a variety of additional features such as querying and reasoning engines. Pellet Reasoner and OntoGraf are two helpful plug-ins used for this research. Pellet is a reasoner plug-in for Protégé providing various functionalities such as logical consistency checking, concept satisfiability, automatic classification and realization while OntoGraf allows users to visualize classification hierarchies and relationships among classes. Projects developed in Protégé can be encoded in various formats such as RDF, OWL and XML schema. In this project, OWL is chosen to code the ontology because it makes the task of reasoning with expressions more tractable.

6.3.1.1. Ontology modelling part

After a literature review with keywords "home healthcare" and "ontology-based healthcare", two existing ontologies have been found which are highly related to parts of the proposed ontology. In general, three components make up the entire ontology. One component of the home healthcare ontology proposed in this thesis related to emotion is inherited from [87]. [87] introduces an ontology based representation of the affective states. It also specifies the complex relations that among the affective states and between them and the other context elements. More precisely, only the class of **State** is selected and reused in the proposed ontology in this work. Another component is extracted from [88] in which authors make an ontology for the mental health, moreover exact factors leading to those mental disorders are presented. Several disorders appeared with high frequencies are selected into our ontology. Generally, the structure of the proposed ontology is shown in figure 32.

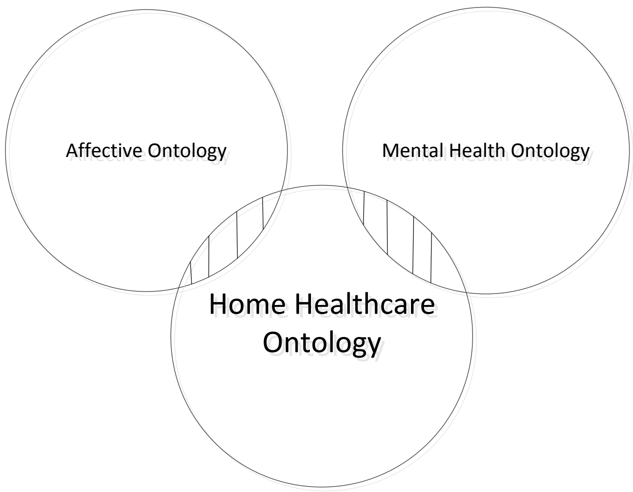

Figure 32. Ontology structure

As revealed in figure 32, two ontologies are reused in this project to reduce the workload and provide interoperability.

Figure 33 illustrates the general taxonomy of a home healthcare ontology which is an example to practically implement the whole process learned before theoretically. More details are shown as a supplement to provide a better understanding for the ontology in the following.

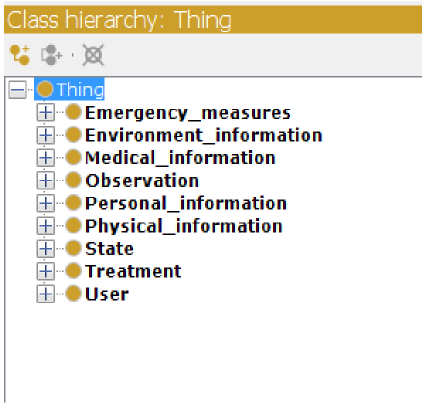

Figure 33. Top level hierarchy of the ontology

Figure 33 is a screenshot of the top level hierarchy of the ontology created in Protégé indicating that nine main classes make up the ontology. More specifically, they are **User**, **Treatment**, **State**, **Physical_information**, **Personal_information**, **Observation**, **Medical_information**, **Environment_information** and **Emergency_measures**. These classes describe the concepts of different domains of interest. Each class implies the conditions that an individual must meet for being part of the class. The names are selected so that people can easily understand what area or domain each class represents from the literal meaning, for example, the class **User** should be a set of people who are involved in the home healthcare system, namely, users. Actually the classes are organized into a superclass-subclass hierarchy. Subclasses are descendants of superclass representing the domain in a more specific way. Next, each superclass is going to be unfold and described in detail. In the following, the breakdown of each superclass will be presented.

- **User**. It is any person who is involved in the home healthcare system. Apparently indicated from the figure 34, five types of users are defined including Patient, Doctor, Nurse, Psychologist and Relative. The clear classification for users brings many benefits. Among all, the major advantage is to prohibit unauthorized users from logging into the system and retrieving personal information. Due to this explicit definition, security for context data should be guaranteed.

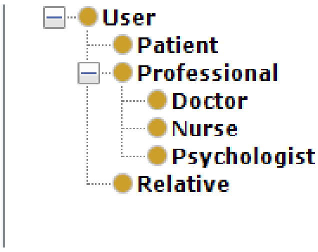

Figure 34. Hierarchy of user

- **Treatment**. Personalized treatment is a rather complicated issue requiring several aspects to be taken into account such as the patient's medical history, the patient's preferences and response to the treatment. Therefore, it is closely related to other classes like patient's profile and observation. In the scenario, five kinds of treatment are offered in order to meet different patients' needs. The detailed classification is showed in figure 35. The **Music_plan** aims to help the patients to relax showing musical preferences. For patients with chronical mental diseases, doctors are able to make different prescriptions according to their specific diseases. For example, **Drug_therapy** is good to cure patients with mental diseases like anxiety, depressant or other general mental diseases who need to take pills. **Group_and_family_therapies** is a kind of therapy for patients with the collaboration from patients' group and family. As to **Psychosurgery**, it means a surgery for seriously ill patients. Apart from those aforementioned treatment means, **Exercise_plan** which contains four specific exercises is suitable for patients who like doing exercises. For example, if a stroke patient adores doing exercises, it is good to let him recover his ability during doing his favorite thing. More types of treatment can meet patients' needs as much as possible.

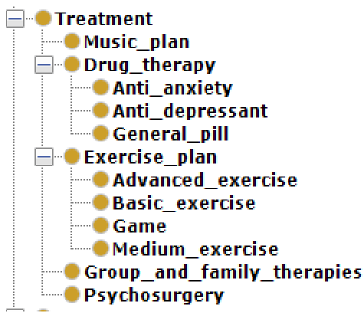

Figure 35. Hierarchy of treatment

- **State**. This class is inherited from [87]. The **Affective** state of a person is regarded as a significant context factor, e.g. a significant change of the affective state can be a sign for an abnormal phenomenon. In [87], the affective state is defined by three subclasses as shown in figure 36: **Basic** affective states are **Angry**, **Disgust**, **Fear**, **Happy**, **Neutral**, **Sad** and **Surprise**, the **Secondary** state expresses **Boredom**, **Curiosity**, **Enthusiasm** and **Stress** and the **Current** state is determined by the cooperation of the **Basic** and the **Secondary** state.

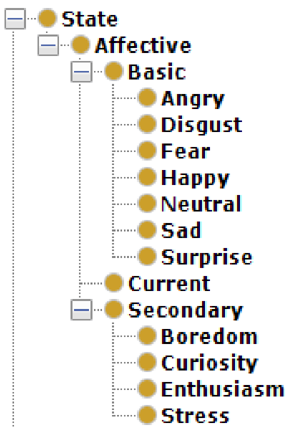

Figure 36. Hierarchy of state

- **Physical_information & Personal_information.** A patient's profile is formed of these two superclasses containing most useful information about the patient and his body health index which is a critical criterion for the doctor to make a diagnosis. **Physical_information** and **Personal_information** demonstrate basic concepts about a patient's body index from different aspects. The entire OntoGraf hierarchy of patient's profile is shown in figure 37.

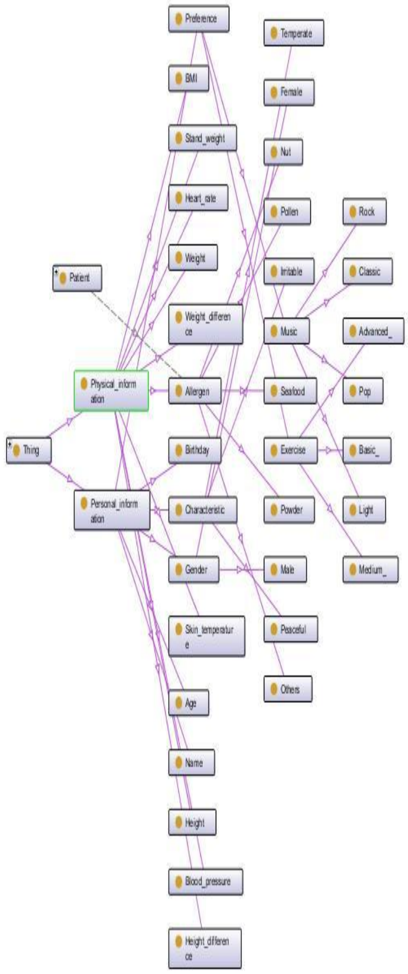

Figure 37. Hierarchy of patient profile

More precisely, figure 38 illustrates the structure of **Personal_information**. It indicates that the class of **Personal_information** would contain subclasses or other concepts to describe personal information like **Age**, **Birthday**, **Gender**, **Name** etc. These concepts are the primary information coming into one's mind when people talk about a person. Besides, they are important for identifying a person. Apart from these concepts, it is worth noting that characteristics and preference are also useful to describe a person. By knowing more personal information, the system could supply more customized services. **Characteristic** is divided into three subclasses which are **Irritable**, **Temperate** and **Peaceful**. These are typical examples of personality expressing three different intensities of individual differences among people in behavior patterns, attitude, emotion or mood in daily life. They are sorted from the strongest one (**Irritable**) to the weakest one (**Peaceful**). In a similar way, the class of Exercise is composed of three subclasses which are **Advanced_**, **Basic_** and **Medium_** representing three activities with different intensity. **Music** is an art of beautiful sound. Here a popular classification for **Music** is adopted which is composed of **Classic**, **Light**, **Pop** and **Rock**. Based on what was already defined for the concepts of **Exercise** and **Music**, it can be found that all of these concepts can meet the requirements for membership of **Preference**. Hence, all those aforementioned concepts should belong to the superclass **Personal_information**.

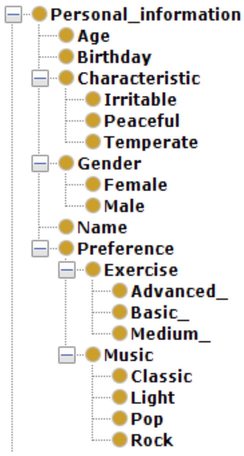

Figure 38. Hierarchy of Personal_information

Physical_information is a concept defining people's body index. **Allergen** is a class expressing the specific antigen that can stimulate an allergic reaction including **Nut**, **Pollen**, **Powder**, **Seafood** and **Others**. **Blood_pressure**, **Heart_rate**, **Height**, **Weight** and **Skin_temperature** are also vital information to maintain and assess a person's health. It is worth mentioning that **Stand_weight** defines the standard weight for individuals which can be calculated through an equation that is going to be presented in the next section. **Height_difference** simply means a value got from **Height** minuses 100.0 while **Weight_difference** is the remainder of taking **Stand_weight** from **Weight**. **BMI** means **Body Mass Index** as a measure of relative weight based on an individual mass and height. More details about **BMI** can be seen in 6.3.1.2.

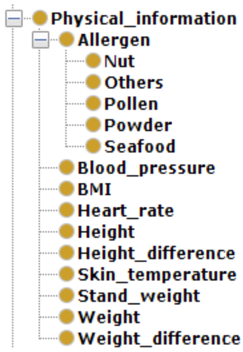

Figure 39. Hierarchy of Physical_information

- **Observation**. The performance results achieved by patients after they take therapies are defined as **Observation**. **Doctors** could use **Observation** as feedbacks in order to change prescriptions accordingly. There are two subclasses: **Effective** and **Ineffective**, referring to the efficiency of the therapy.

Figure 40. Hierarchy of Observation

- **Medical_information**. The class of **Medical_information** contains all diseases which are possible to be diagnosed in the home healthcare scenario. In general, it is structured into two subclasses: **Physical** and **Psychologic**. In the **Physical** class, three main characteristics are listed: **Cardiac**, **Elderly**, and **Stroke**. As discussed earlier, the class of **Psychologic** is inherited from a complete ontology for studying the mental diseases and their causes in [88]. However, only several common mental diseases are selected and used in this ontology including **Anxiety_disorder** and **Personality_disorder**. In addition, **Anxiety_disorder** is composed of four more specific disorders: **Genralized_anxiety_disorder**, **Panic_disorder**, **Social_disorder** and **Specific_phobia**.

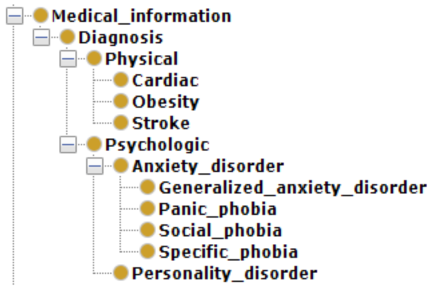

Figure 41. Hierarchy of Medical_information

- **Environment_information**. It makes a taxonomy for concepts possibly existing in a normal environment including **Brightness**, **Humidity**, **Location**, **Temperature** and **Time**. As to the location, because the system is limited to home, so the location is divided into several areas distinguished by name and functionality, namely, they are **Bedroom**, **Dining_room**, **Kitchen** and **Toilet**. Besides, the intensity of brightness and humidity is set to different degrees as **Bright** and **Dark**.

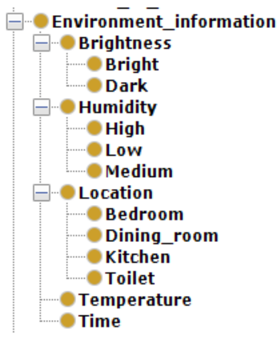

Figure 42. Hierarchy of Environment_information

- **Emergency_measures**. This term defines all possible measures taken in a home healthcare system to cope with the abnormal phenomenon. **Call_Doctor** means a reaction of calling a doctor to inform the emergency while **Text_to_Relatives** defines a measure of sending text to relatives for asking help. What is more, the term of **Rise_Alarm** expresses another way to deal with emergency which is rising an alarm locally to draw people's attention nearby.

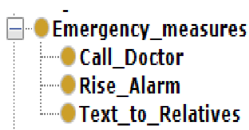

Figure 43. Hierarchy of Emergency_measures

The classes themselves are not sufficient to provide enough information. Relationships between the classes are needed to form an ontology. They are binary connections defining the inter-relationships between classes or individuals. Relationships allow developers to depict the internal structure of concepts and also refer to property. They also allow people to determine which class it describes and what domain or range it specifies. The tab called **Object Properties** which can be seen in figure 44 in Protégé allows users to graphically edit the relationships.

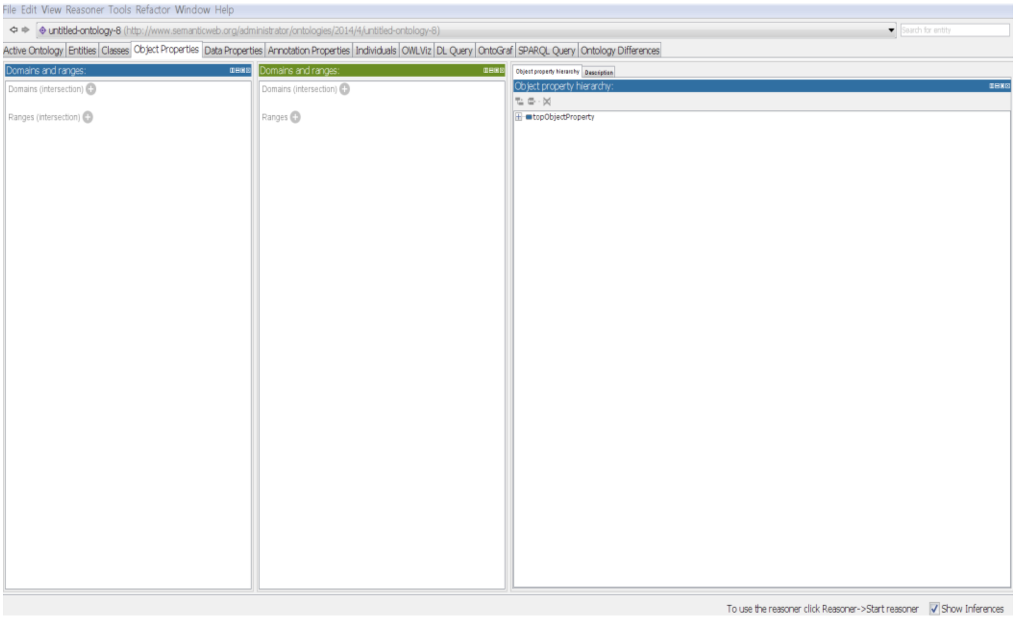

Figure 44. The tab of Object Properties in Protégé

The general hierarchy of object properties can be seen in figure 45.

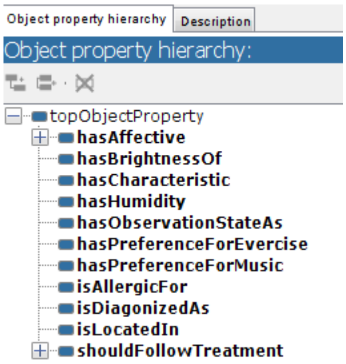

Figure 45. The hierarchy of object properties

The following properties are defined for the home healthcare ontology.

- **hasAffective**. **Patient** ⟶ **State**

- **hasBasicStateAs**. **Patient** ⟶ **Basic**
- **hasSecondaryState**. **Patient** ⟶ **Secondary**
- **hasBrightnessOf**. **Location** ⟶ **Brightness**
- **hasCharacteristic**. **Patient** ⟶ **Characteristic**
- **hasHumidity**. **Location** ⟶ **Humidity**
- **hasObservationStateAs**. **Treatment** ⟶ **Observation**
- **hasPreferenceForExercise**. **Patient** ⟶ **Exercise**
- **hasPreferenceForMusic**. **Patient** ⟶ **Music**
- **isAllergicFor**. **Patient** ⟶ **Allergen**
- **isDiagnosedAs**. **Patient** ⟶ **Medical_information**
- **isLocatedIn**. **Patient** ⟶ **Location**
- **shouldFollowTreatment**. **Patient** ⟶ **Treatment**
- **shouldFollowDrugTherapy**. **Patient** ⟶ **DrugTherapy**
- **shouldFollowExercisePlan**. **Patient** ⟶ **ExercisePlan**
- **hasBMI. Patient** ⟶ **BMI**

Above all, they are all object properties linking different classes. The general schema is the class in the left side of the arrow represents the domain of the object property which is the class that the relationship is attached to. For example, the object property of **hasAffective** links the class of **Patient** with the class of **State** which specifies that all the possibility of affective states that **Patient** could have must be in the class of **State**. In similarity, the functions of other relationships can be simply understood as their names imply since their literal meaning is quite straightforward.

Finally, a figure displaying the whole ontology model including classes and relationships which is provided by OntoGraf is presented in figure 46.

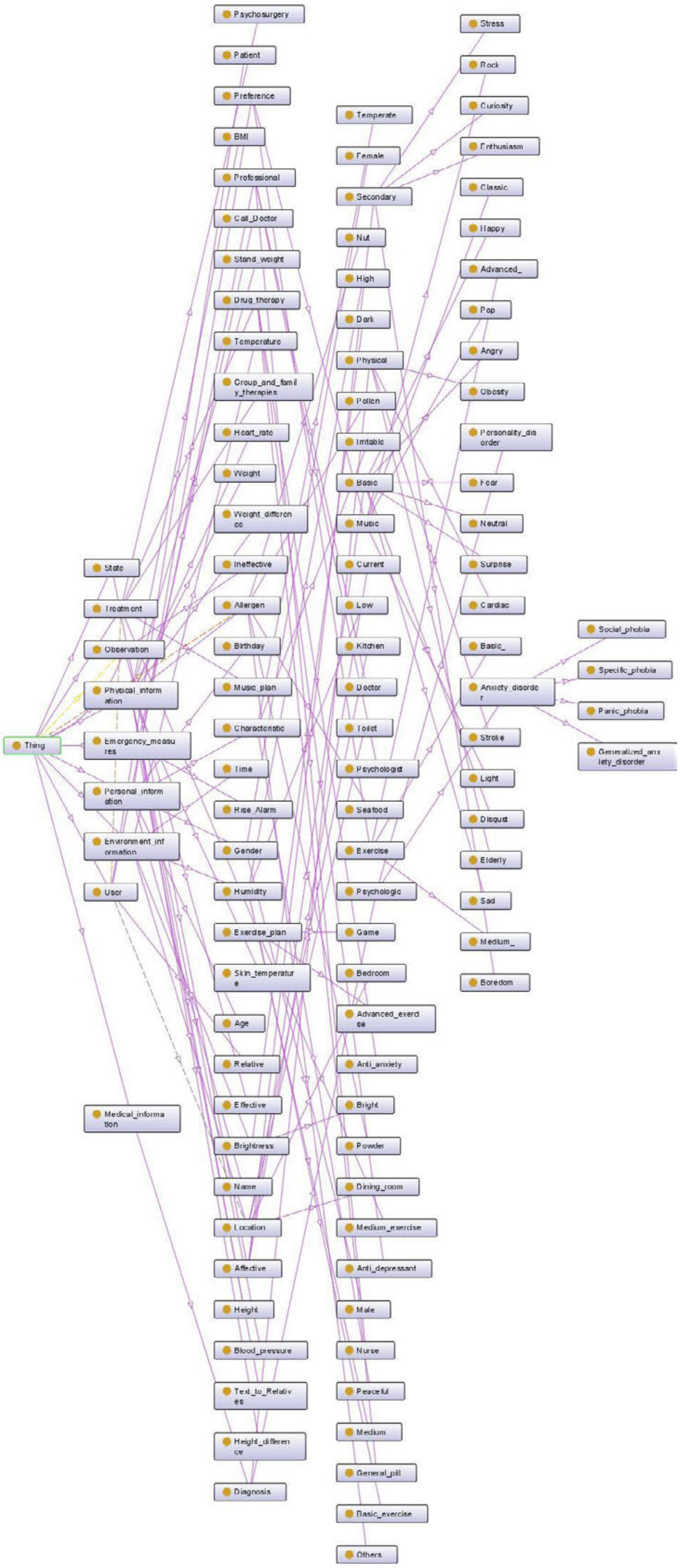

Figure 46. OntoGraf hierarchy display for home healthcare ontology

6.3.1.2. Ontology reasoning part

Implicit context as deduced information from explicit information is obtained from the ontology reasoning. First of all, typical OWL ontology reasoning rules are employed such as ***subClassOf***, ***subPropertyOf***, ***disjointWith, transitive, symetric*** and ***inverseOf*** etc. For example, it can be inferred that an individual *Jack* belongs to ***Patient*** is different with *David* who belongs to ***Doctor*** if the property for **Patient** and **Doctor** is set as ***disjointWith***. But only using OWL expressivity can just implement part of the conditions,

SWRL (**Semantic Web Rule Language**) as the extension of OWL is adopted to define user-defined rules for the ontology. It allows users to write Horn-like rules expressed in terms of OWL concepts to reason about the ontology based on OWL. The rules can be used to infer new knowledge and high level context from the existing OWL knowledge bases. A **SWRL** rule contains an antecedent part and a consequent part. Both the antecedent and consequent consist of positive conjunctions of atoms. Informally, a SWRL rule may be read as: if all the atoms in the antecedent are true, then the consequent must also be true.

Besides, **SWRL** can use arithmetic operators and can compute the desired behavior based on the context of the individual which could depend on a dynamic environment with multiple components.

In this way, rules for the combined context and inferred context like treatment, affective state and BMI etc. can be defined. To edit the user-defined rules, the SWRL rules tab as figure 47 could be explored.

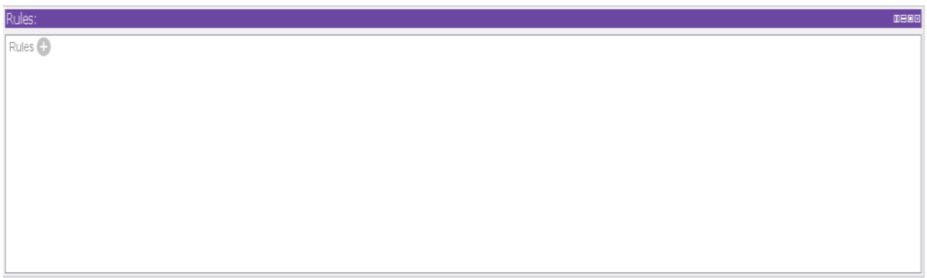

Figure 47. The SWRL rules tab

For instance, the inference of BMI can be calculated by applying the height and weight of a person as follows:

$$Stand\_weight=(Height-100.0)\times 0.8 \tag{2}$$

$$BMI=\frac{Weight-Stand\_weight}{Stand\_weight}\times 100.0 \tag{3}$$

The rules which have to be specified in SWRL for this calculation are explained step by step in the following.

Height_difference Rule:

Height(?y), User(?x), hasHeightOf(?x, ?y), subtract(?z, ?y, 100.0) -> Height_difference(?z)

The meaning of this sentence is if the height value of a person (x) is y, then by subtracting 100.0 from y the height difference (z) can be obtained.

Stand_weight Rule:

Height_difference(?y), User(?x), hasHeightDifferenceOf(?x, ?y), multiply(?z, ?y, 0.8) -> Stand_weight(?z)

The rule explains that the stand_weight (z) of a person can be obtained by the multiplication of height difference and 0.8.

Weight_difference Rule:

User(?x), Stand_weight(?y), Weight(?z), hasStandardWeightOf(?x, ?y), hasWeightOf(?x, ?z), subtract(?a, ?z, ?y) -> Weight_difference(?a)

This rule specifies the way to get the weight difference of a person which simply equals the difference between weight and stand_weight.

BMI Rule:

User(?x),Stand_weight(?b),Weight_difference(?y), hasStandardWeightOf(?x, ?b), hasWeightDifferenceOf(?x, ?y), divide(?a, ?z, ?b), multiply(?z, ?y, 100) -> BMI(?a)

The sentence above fulfills the function of equation 3. The value of a person's BMI is calculated by this rule which takes many context parameters into account.

Also user-defined rules for treatment advice can be specified in SWRL.

Basic_exercise Rule:

User(?x), hasBMI(?x,?y), swrlb:greaterThan(?y,?30)->Patient(?x)

A simple rule is defined to determine if a person is attributed to the class of patient by simply taking BMI into consideration. If the BMI value of a person is greater than the threshold (here is 30), the person can be inferred as a patient.

```
Patient(?x),isDiagonosedAs(?x,?y),Stroke(?y),hasPreferenceForE
xercise(?x,?z),Basic_degree(?z),shouldFollowExercisePlan(?x,?w
)->Basic_exercise(?w)
```

The sentence above shows the procedure to automatically generate a treatment which is the basic exercise for a stroke patient by taking his preferences for the exercise and its intensity.

Similarly, other inference rules can be specified like the format specified above in SWRL. The range of SWRL expression is broader than the existing ontology language. With the flexibility and feasibility supplied by SWRL, users can define any rule as they wish.

6.3.2. Validation

The final stage of developing an ontology is validation. First, the logical consistency between different classes should be validated. Afterwards, the predefined competency questions are reminded to check if the ontology-based model is complete and expressive enough to answer all of them. If more relevant information in response to ontology questions is provided, it can get the conclusion that the ontology is able to meet and support the scope and purpose in which it is applied [89].

The focus of validation stage in this thesis is checking the logical consistency followed by a few simple tests for the proposed ontology.

The software adopted to develop the ontological model, Protégé 4.3, continuously provides the Pellet ontology reasoner as a plug-in and is able to check for the existing inconsistencies in the ontology. The reasoner is able to perform several services such as permitting automatic classification, finding out if classes are consistent or identifying subsumption relationships between classes. Pellet can automatically compute the classification taxonomy and check the consistency of ontology. Once the reasoner completes the process, the Protégé function "DL

Query" and "SPARQL" are used to make queries for information about the ontology. In Protégé 4.3, asserted hierarchy refers to the class hierarchy which is manually constructed while the class hierarchy automatically computed by the reasoner is called inferred hierarchy. The concrete procedures to validate the ontology are as follows:

- To start the Pellet reasoner, invoke the "Start Reasoner" in the Reasoner drop down menu.
- Then all the classes and relationships are computed by reasoner.
- If any inconsistency is found, the corresponding classes or properties are going to be changed in red alarming that mistakes have occurred.

After performing the aforementioned actions in Protégé, the result as figure 48 shows that there are no inconsistencies in the proposed ontology because no class is in red.

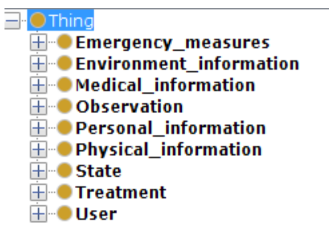

Figure 48. The result after executing Pellet reasoner

The tab of SPARQL [90] in Protégé can help to make some examinations for the proposed ontology. The default interface of SPARQL can be seen in figure 49.

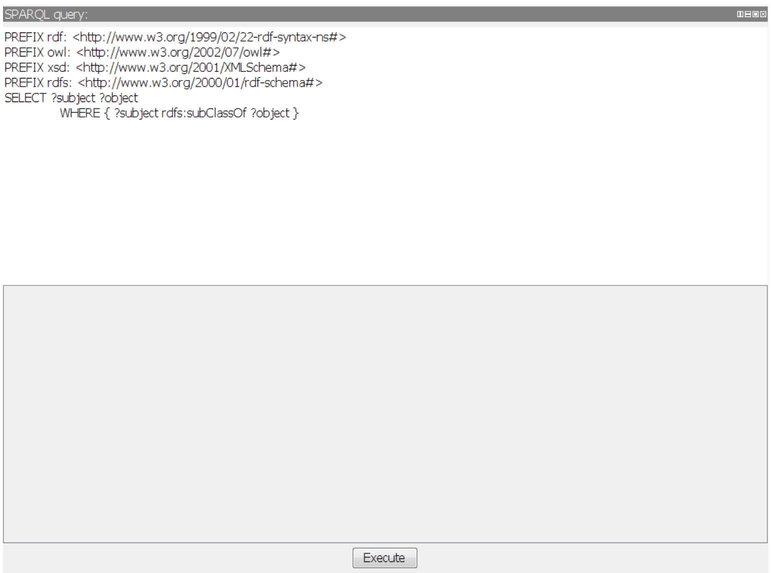

Figure 49. The SPARQL tab

In this tab, queries can be made and coded in SPARQL query language. The SPARQL query language is closely related to the SPARQL protocol for RDF and the SPARQL query results XML format specification. In the following, a few examples about how to make queries in SPARQL are listed.

To query the classes linked with an object property like **hasAffective**, the code is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX test:<http://www.semanticweb.org/administrator/ontologies/2014/4/untitled-ontology-8#>

SELECT ?domain ?range

WHERE {test:hasAffective  rdfs:domain ?domain; rdfs:range ?range }
```

After execution, the results containing all the classes linked with the relationship can be seen as figure 50:

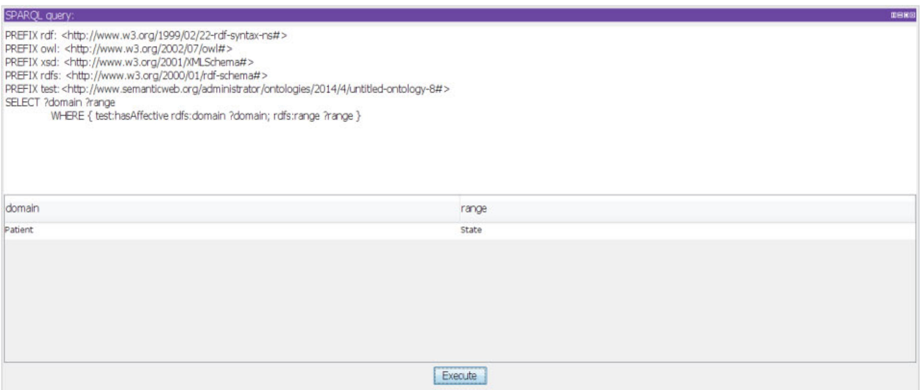

Figure 50. Query for all classes related to *hasAffective*

As depicted from the figure 50, the SPARQL query tab can output all the classes which are linked with the relationship (here the example is *hasAffective* links Patient to State). Similarly, details for other object relationships can be checked and queried via the SPARQL query tab.

In the following, more queries for the proposed ontology are listed. Figure 51 shows the result of a query for the superclass of a class called Patient in the proposed ontology. The class User is found and output as the superclass of Patient. If changing the class Patient to other subclasses, those superclasses corresponding to the specific subclasses can be located as well.

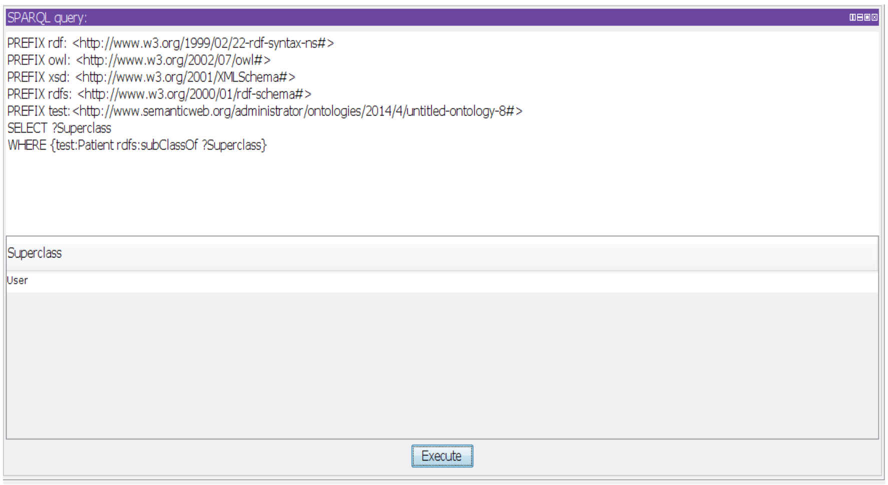

Figure 51. The query for the superclass of Patient in the proposed ontology

Figure 52 is another query example showing the way to make queries about the subclasses of a specific class. More precisely, from the figure it can be obtained that three classes including Patient, Relative and Professional are the subclasses of the class User.

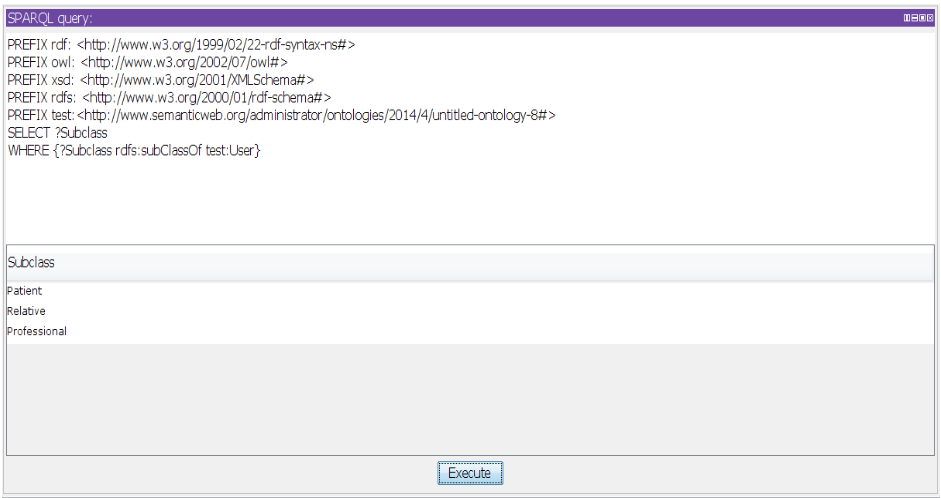

Figure 52. The query for the subclasses of Patient in the proposed ontology

The SPARQL tab provides users the interface and the language to consult any information about the ontology. Details about the ontology in many aspects can be queried and consumed e.g. concepts, relationships, the total number of classes etc.

7 Conclusion and discussion

The current development of IoT is suffering from the lack of some indispensable capabilities, such as interoperability, context awareness and security & privacy, among which context awareness is becoming more accepted by academia and industry as an ideal solution. Context awareness facilitates any type of system to provide customized services without human intervention. Aiming at fusing context awareness into IoT, a preliminary research has been carried out in this thesis. A thorough survey of the existing context modelling and context reasoning techniques is conducted highlighting the ontology-based modelling and reasoning techniques. Compared with other techniques, ontology is regarded as the most promising technique to construct and infer context for its expressivity, formality, interoperability and the support of reasoning. This thesis also introduces several popular context awareness frameworks e.g. Cobra, Context Toolkit, Gaia, SIM, Aura, MoCA and HCoM, based on which a new ontology-based context awareness framework for IoT is proposed. The novelty of the proposed framework is that it takes some aspects into account such as the integration of ontology with IoT, security & privacy and the storage of data which are not implemented in other frameworks. To utilize the ontology-based modelling and reasoning techniques, the home healthcare area is selected as a test environment. First of all, a new ontology developing method is extracted from the TOVE approach and it modifies TOVE into four steps referred to as Specification & Conceptualization, Competency Formulation, Implementation and Validation & Documentation to adapt to the home healthcare area which provides a clear and detailed guidance to create an ontology. Following the proposed ontology developing method, an ontology-based model for a home healthcare scenario is created using a software tool called Protégé. Details about the internal hierarchy of the ontology and descriptions for its concrete concepts are provided. Finally, this proposed ontology is validated. The result shows that the proposed ontology has no inconsistencies, and queries for classes and relationships can be achieved via SPARQL.

However, it is worth noting that there are still a few limitations and shortages to overcome. Future work could be put on aspects as follows:

- The proposed ontology in this thesis is simple and not complete because it is created by a single person. Future work should aggregate various viewpoints of multiple individuals to make the ontology more complete and comprehensive.
- Literature research is the main resource to collect data and help the author to understand the current development situation of the area. So the knowledge base is limited to some extent. More helpful resources such as practice experience, real projects or scenarios should be used to enrich the knowledge base of the ontology.
- The proposed ontology based context awareness framework is only proved at a theoretical level. Potential work should test the proposed framework in real projects.
- The internal validation for the proposed ontology is not complete as the competency questions are not documented in the thesis. Further validation could be made in the future from two aspects: 1. Add all the predefined competency questions and check if the proposed ontology is expressive and complete to provide sufficient information to answer those questions. 2. External validation also needs to be accomplished.
- Rules existing in the proposed ontology could be refined in terms of rationality and completeness.
- The proposed ontology model is developed in Protégé and written in OWL. Future research should also involve annotating the ontology using other modelling languages.

References

[1] P. Guillemin and P. Friess, "Internet of things strategic research roadmap," The Cluster of European Research Projects, Tech. Rep., September 2009, http://www.internet-of-things-research.eu/pdf/IoT Cluster Strategic Research Agenda 2009.pdf [Accessed on: 2011-08-15].

[2] L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805.

[3] D. Guinard, "Towards the web of things: Web mashups for embedded devices," in In MEM 2009 in Proceedings of WWW 2009. ACM, 2009.

[4] K. Ashton, "That internet of things thing in the real world, things matter more than ideas," RFID Journal, June 2009, http://www.rfidjournal.com/article/print/4986 [Accessed on: 2012-07-30].

[5] T. Lu and W. Neng, "Future internet: The internet of things," in 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 5, August 2010, pp. V5–376–V5–380. [Online]. Available: http://dx.doi.org/10.1109/ICACTE.2010.5579543

[6] European Commission, "Internet of things in 2020 road map for the future," Working Group RFID of the ETP EPOSS, Tech. Rep., May 2008, http://ec.europa.eu/information society/policy/rfid/documents/iotprague2009.pdf [Accessed on: 2011-06-12].

[7] IEEE 802.15 WPAN™ Task Group 4, http://www.ieee802.org/15/

[8] W3C Community and Business Groups, http://www.w3.org/community/wot/

[9] V. Issarny, M. Caporuscio, and N. Georgantas: "A Perspective on the Future of Middleware-Based Software Engineering," Future of Software Engineering 2007, L. Briand and A. Wolf (eds.), IEEE-CS Press, 2007.

[10] K. Lund, T. Hafse, and F. T. Johnsen. A survey of middleware with focus on application in network based defence. FFI report 2007/02683, 2007.

[11] Dey, A. K., & Abowd, G. D. (2000b). Towards a better understanding of context and context- awareness. Proceedings of the What, Who, Where, When, and How of Context-Awareness Workshop, CHI 2000 Conference on Human Factors in Computer Systems. New York: ACM.

[12] Schilit, B. (1995). System architecture for context-aware mobile computing. Unpublished doctoral dissertation, Columbia University, New York.

[13] Brown, P. J., Bovey, J. D., & Chen, X. (1997). Context-aware applications: From the laboratory to the marketplace. IEEE Personal Communications, 4(5), 58–64.

[14] Pascoe, J., Ryan, N. S., & Morse, D. R. (1998). Human–computer–giraffe Interaction: HCI in the field. Proceedings of the Workshop on Human Computer Interaction with Mobile Devices. GIST Technical Report G98–1. Glasgow, Scotland: University of Glasgow.

[15] Franklin, D., & Flaschbart, J. (1998). All gadget and no representation makes jack a dull environment. Proceedings of the AAAI 98 Spring Symposium on Intelligent Environments. Menlo Park, CA: AAAI Press.

[16] Ward, A., Jones, A., & Hopper, A. (1997). A new location technique for the active office. IEEE Personal Communications, 4(5), 42–47.

[17] Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). Towards situated computing. Proceedings of the 1st International Symposium on Wearable Computers (ISWC 97). Los Alamitos, CA: IEEE.

[18] Chen, H.: An Intelligent Broker Architecture for Pervasive Context-Aware Systems. Ph.D. Thesis, University of Maryland, Baltimore County (2004)

[19] Henricksen, K.: A Framework for Context-Aware Pervasive Computing Applications.

Ph.D. Thesis, University of Queensland, Queensland, Queensland (2003)

[20] A.K.Dey, et al. “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of context-Aware Applications”, Human-Computer Interaction Journal, Vol. 16(2-4), pp. 97-166, 2001.

[21] ZIMMERMANN, A., LORENZ, A., AND OPPERMANN, R. 2007. An operational definition of context. In Proceedings of the 6th International and Interdisciplinary Conference on Modeling and using Context (CONTEXT07), 558–571.

[22] Kim, J., Son, J., Baik, D.: CA 5W1H onto: ontological context-aware model based on 5W1H. International Journal of Distributed Sensor Networks (2012) .

[23] D. Zhang, H. Huang, C. Lai et al., "Survey on context-awareness in ubiquitous media", Multimedia Tools and Applications, Springer Netherlands,pp. 1-33, 2011.

[24] Lee, K. C., Kim, J. H., Lee, J. H., & Lee, K. M. (2007). Implementation of ontology based context-awareness framework for ubiquitous environment. In Proceedings of the 2007 international conference on multimedia and ubiquitous, engineering (pp.278–282).

[25] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in Proc. 1st international symposium on Handheld and Ubiquitous Computing, ser. HUC '99. London, UK: Springer-Verlag, 1999, pp. 304–307. [Online]. Available: http://dl.acm.org/citation.cfm?id=647985.743843

[26] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," 2013.

[27] Karen Henricksen, et al. "Modeling Context Information in Pervasive Computing Systems", Pervasive 2002.

[28] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," Network, IEEE, vol. 8, no. 5, pp. 22 –32, sep/oct 1994. [Online]. Available: http://dx.doi.org/10.1109/65.313011

[29] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in Proc. 1st international symposium on Handheld and Ubiquitous Computing, ser. HUC '99. London, UK: Springer-Verlag, 1999, pp. 304–307. [Online]. Available: http://dl.acm.org/citation.cfm?id=647985.743843

[30] Hull, R., Neaves, P., Bedford-Roberts, J. Towards Situated Computing. 1st International Symposium on Wearable Computers (1997) 146-153

[31] Pascoe, J. Adding Generic Contextual Capabilities to Wearable Computers. 2nd International Symposium on Wearable Computers (1998) 92-99

[32] Dey, A.K. Context-Aware Computing: The CyberDesk Project. AAAI 1998 Spring Symposium on Intelligent Environments, Technical Report SS-98-02 (1998) 51-54

[33] Dey, A.K., Abowd, G.D., Wood, A. CyberDesk: A Framework for Providing Self-Integrating Context-Aware Services. Knowledge-Based Systems, 11 (1999) 3-13

[34] Salber, D., Dey, A.K., Abowd, G.D. Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm. Georgia Tech GVU Technical Report GIT-GVU-98-01 (1998)

[35] Brown, P.J. Triggering Information by Context. Personal Technologies, 2(1) (1998) 1-9

[36] Ryan, N. Mobile Computing in a Fieldwork Environment: Metadata Elements. Project working document, version 0.2 (1997)

[37] L. Barkhuus, L. Barkhuus, and A. Dey, "Is context-aware computing taking control away from the user three levels of interactivity examined," in In Proc. Ubicomp 2003. Springer, 2003, pp. 149–156. [Online]. Available: http://www.itu.dk/people/barkhuus/barkhuusubicomp.pdf

[38] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," 2013.

[39] G. Hynes, V. Reynolds, and M. Hauswirth, "A context lifecycle for web-based context management services," in Smart Sensing and Context, ser. Lecture Notes in Computer Science, P. Barnaghi, K. Moessner, M. Presser, and S. Meissner, Eds. Springer Berlin/ Heidelberg, 2009, vol. 5741, pp. 51–65. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04471-7 5

[40] M. Peterson and E. Pierre, "Snias vision for information life cycle management (ilm)," in Storage Networking World. Computer World, 2004.

[41] AIIM, “What is enterprise content management (ecm)?” February 2009, http://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx [Accessed on: 2012-06-20].

[42] E. Hayden, “Data lifecycle management model shows risks and integrated data flow,” in Information Security Magazine, July 2008.

[43] A. N. Shulsky and G. J. Schmitt, Silent Warfare: Understanding the World of Intelligence, 3rd ed. Potomac Books Inc, May 2002.

[44] J. R. Boyd, “A discourse on winning and losing.” Unpublished set of briefing slides available at Air University Library, Maxwell AFB, Alabama, 1987, http://www.ausairpower.net/JRB/intro.pdf [Accessed: 2011-12-18].

[45] M. Chantzara and M. Anagnostou, “Evaluation and selection of context information,” in In: Second International Workshop on Modeling and Retrieval of Context, Edinburgh, 2005. [Online]. Available: http://ceur-ws.org/Vol-146/paper7.pdf

[46] A. Ferscha, S. Vogl, and W. Beer, “Context sensing, aggregation, representation and exploitation in wireless networks,” Scalable Computing: Practice and Experience, vol. 6, no. 2, p. 7181, 2005. [Online]. Available: http://www.scpe.org/index.php/scpe/article/view/327/17

[47] G. Hynes, V. Reynolds, and M. Hauswirth, “A context lifecycle for web-based context management services,” in Smart Sensing and Context, ser. Lecture Notes in Computer Science, P. Barnaghi, K. Moessner, M. Presser, and S. Meissner, Eds. Springer Berlin/ Heidelberg, 2009, vol. 5741, pp. 51–65. [Online]. Available:http://dx.doi.org/10.1007/978-3-642-04471-7 5

[48] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context aware systems,” Int. J. Ad Hoc Ubiquitous Comput., vol. 2, no. 4, pp. 263–277, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1504/IJAHUC.2007.014070

[49] http://en.wikipedia.org/wiki/List_of_sensors

[50] K. Henricksen, “A framework for context-aware pervasive computing applications,” Computer Science, School of Information Technology and Electrical Engineering, The University of Queensland, September 2003, http://henricksen.id.au/publications/phd-thesis.pdf [Accessed on: 2012-01-05].

[51] S. Yanwei, Z. Guangzhou, and P. Haitao, “Research on the context model of intelligent interaction system in the internet of things,”in IT in Medicine and Education (ITME), 2011 International Symposium on, vol. 2, dec. 2011, pp. 379 –382. [Online]. Available: http://dx.doi.org/10.1109/ITiME.2011.6132129

[52] T. Strang and C. Linnhoff-Popien, “A context modeling survey,” in In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England, 2004. [Online]. Available: http://elib.dlr.de/7444/1/ Ubicomp2004ContextWSCameraReadyVersion.pdf

[53] Schilit B, Adams N, Want R. Context-aware computing applications. In: First International Workshop on Mobile Computing Systems and Applications, 1994; 85–90

[54] HELD, A., BUCHHOLZ, S., AND SCHILL, A. Modeling of context information for pervasive computing applications. In Proceedings of SCI2002/ISAS 2002 (2002).

[55] K. Henricksen, J. Indulska, and T. McFadden. Modelling context information with orm. In Proceedings of On the Move to Meaningful Internet Systems 2005: OTM Workshops, pages 626–635, 2005.

[56] CHEVERST, K., MITCHELL, K., AND DAVIES, N. 1998. Design of an object model for a context sensitive tourist guide. In Proceedings of the Conference on Interactive Applications of Mobile Computing (IMB ’98, Rostock, Germany, Nov.).

[57] Xu C, Cheung SC (2005) Inconsistency detection and resolution for context-aware middleware support. In: ESEC/FSE-13: proceedings of the 10th European software engineering conference. ACM, Lisbon, pp 336–345

[58] Bradley N, Dunlop M (2003) Towards a multidisciplinary user-centric design framework for context-aware applications. In: Proceedings of the 1st UK-UbiNet workshop. Springer-Verlag, London, pp 25–26

[59] G. Castelli, A. Rossi, M. Mamei, F. Zambonelli, A simple model and infrastructure for context-aware browsing of the world, in: Proceedings of the 5th IEEE Conference on Pervasive Computing and Communications, IEEE Computer Society, 2007.

[60] Dongpyo Hong, Hedda R. Schmidtke, and Woontack Woo. Linking context modelling and contextual reasoning. In 4th International Workshop on Modelling and Reasoning in Context, Roskilde, Denmark, 2007 2007.

[61] R. Studer, V. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," Data & Knowledge Engineering, vol. 25, no. 12, pp. 161 – 197, 1998. [Online]. Available: http://dx.doi.org/10.1016/S0169-023X(97)00056-6

[62] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," Stanford University, Stanford, CA, Tech. Rep., March 2001, http://protege.stanford.edu/publications/ontology development/ontology101.pdf [Accessed on: 2011-12-15].

[63] M. Uschold, M. Gruninger, Ontologies: principles, methods, and applications, Knowledge Engineering Review 11 (2): (1996) 93-155.

[64] Korpipää, P., Mäntyjärvi, J., Kela, J., Keränen, H., and Malm, E. (2003). Managing Context Information in Mobile Devices. *IEEE Pervasive Computing*, 2(3) :42–51.

[65] A. Bikakis, T. Patkos, G. Antoniou, , and D. Plexousaki, "A survey of semantics-based approaches for context reasoning in ambient intelligence," in Ambient Intelligence 2007 Workshops, M. M, F. A, and A. E, Eds., vol. 11. SPRINGER-VERLAG BERLIN, 2008. [Online]. Available: http://www.csd.uoc.gr/~bikakis/pubs/survey-ami07.pdf

[66] Mamei M, Nagpal R (2007) Macro programming through bayesian networks: distributed inference and anomaly detection. In: Proceedings of the 5th annual IEEE international conference on pervasive computing and communications. IEEE, Washington, pp 87–96

[67] Kofod Petersen, A, Mikalsen M (2005) Context: representation and reasoning: representing and reasoning about context in a mobile environment. Rev Intell Artif 19(3):479–498

[68] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, “A middleware infrastructure for active spaces,” IEEE Pervasive Computing, vol. 1, no. 4, pp. 74–83, Oct. 2002.

[69] Wang XH, Zhang DQ, Gu T, Pung HK (2004) Ontology based context modeling and reasoning using owl. In: Proceedings of the 2nd IEEE annual conference on pervasive computing and communications workshops. IEEE, Orlando, pp 18–22

[70] Bikakis A, Patkos T, Antoniou G, Plexousakis D (2007) A survey of semantics-based approaches for context reasoning in ambient intelligence. In: Proceedings of the workshop artificial intelligence methods for ambient intelligence. Springer, London, pp 15–24

[71] Dempster AP (1968) A generalization of bayesian inference. J R Stat Soc Ser B 30:205–247

[72] H. Chen, T. Finin, A. Joshi, L. Kagal, F. Perich, and D. Chakraborty, “Intelligent agents meet the semantic web in smart spaces,” IEE Internet Computing, vol. 8, no. 6, pp. 69 – 79, nov.-dec. 2004. [Online]. Available: http://dx.doi.org/10.1109/MIC.2004.66

[73] A. K. Dey, G. D. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,” Hum.-Comput. Interact., vol. 16, pp.97–166, December 2001. [Online]. Available: http://dx.doi.org/10.1207/S15327051HCI16234 02

[74] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, “A middleware infrastructure for active spaces,” IEEE Pervasive Computing, vol. 1, no. 4, pp. 74–83, Oct. 2002. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2002.1158281

[75] M. Román and R.H. Campbell, “GAIA: Enabling Active Spaces,” Proc. 9th SIGOPS European Workshop, ACM Press, New York, 2000, pp. 229–234.

[76] S.-H. Baek, E.-C. Choi, J.-D. Huh, and K.-R. Park, “Sensor information management mechanism for context-aware service in ubiquitous home,” IEEE Trans. Consum. Electron., vol. 53, no. 4, pp. 1393 –1400, nov. 2007. [Online]. Available: http://dx.doi.org/10.1109/TCE.2007.4429229

[77] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, “Project aura: Toward distraction-free pervasive computing,” IEEE Pervasive Computing, vol. 1, no. 2, pp. 22–31, Apr. 2002. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2002.1012334

[78] R. de Rocha and M. Endler, “Middleware: Context management in heterogeneous, evolving ubiquitous environments,” Distributed Systems Online, IEEE, vol. 7, no. 4, p. 1, april 2006. [Online]. Available: http://dx.doi.org/10.1109/MDSO.2006.28

[79] D. Ejigu, M. Scuturici, and L. Brunie, “Semantic approach to context management and reasoning in ubiquitous context-aware systems,” in Digital Information Management, 2007. ICDIM ’07. 2nd International Conference on, vol. 1, oct. 2007, pp. 500 –505. [Online]. Available: http://dx.doi.org/10.1109/ICDIM.2007.4444272

[80] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, “Towards a Better Understanding of Context and Context-Awareness,” in Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, 1999, pp. 304–307.

[81] V. Borges and W. Jeberson, “Survey of Context Information Fusion for Sensor Networks based Ubiquitous Systems”, in J. Sens. Actuator Netw., 2013.

[82] Ruth March, “Delivering on the promise of personalized healthcare,” *Personalized Medicine*, vol.7, no.3, pp.327-337, May 2010.

[83] Ali Rammal, Sylvie Trouilhet, Nicolas Singer, Jean-Marie Pécatte,“An Adaptive System for Home Monitoring Using a Multiagent Classification of Patterns International Journal of Telemedicine and Applications , 2008.

[84] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.

[85] Pinto H.S., Martins J.P., 2004. Ontologies: How can they be built? Journal of Knowledge Informatics System, 6, 441-46.

[86] Gruber, T.R., 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies,* 43(5-6), 907-928.

[87] Benta, K.L., Rarău, A., Cremene, M.: Ontology Based Affective Context Representation. In: Proceedings of the 2007 Euro American conference on Telematics and information systems (EATIS 2007), Faro, Portugal (2007)

[88] Hadzic M, Chen M, Dillon T S. Towards the mental health ontology[C]//Bioinformatics and Biomedicine, 2008. BIBM'08. IEEE International Conference on. IEEE, 2008: 284-288.

[89] Gruninger, M., 1996. Designing and evaluating generic ontologies. In Proceedings of ECAI96's workshop on ontological engineering, Budapest, Hungary, 53–64.

[90] http://www.w3.org/TR/rdf-sparql-query/