Politecnico di Milano
Scuola di Ingegneria dell'Informazione

Polo territoriale di Como
Master of Science in Computer Engineering

Activities of Daily Living Recognition through Ontological Reasoning

Supervisor: Prof. Sara Comai
Assistant Supervisor: Eng. Andrea Masciadri

Master Graduation Thesis by: Samuele Galli
Student Id. number: 10347923

Academic Year 2017-2018

"The truth.
It is a beautiful and terrible thing,
and should therefore be treated with caution."
(Albus Silente)

Sommario

Con la continua crescita dell'età media della popolazione mondiale è nostro dovere ricercare costantemente una soluzione efficace per soddisfare le future ed inevitabili richieste di intervento e di aiuto. La sfida principale è quella di allungare il più possibile il tempo che un anziano o una persona fragile passa all'interno della sua abitazione in maniera indipendente, prima di effettuare una richiesta di ospedalizzazione. Questo tema è sensibile tra i ricercatori di tutto il mondo e, con il passare degli anni, sono stati proposti molti approcci differenti per risolvere il problema. Tecnologie sempre più avanzate di scienze come Pervasive System o Internet of Things (IoT) hanno permesso lo sviluppo di sistemi di Ambient Assisted Living (AAL) affidabili e montati all'interno di Smart Homes, sfruttando Home Automation (HA) per garantire tecnologie assistive. In questo progetto di tesi viene presentato un approccio ontologico che è in grado di dare supporto ai metodi non supervisionati presenti nella letteratura riguardante il riconoscimento delle attività giornaliere. I passi più importanti durante lo sviluppo del progetto sono stati l'analisi, il design e l'applicazione pratica di un'ontologia, un dizionario semantico, nel quale sono state definite classi, sottoclassi, individui e relazioni semantiche che compongono un nostro dominio personale riguardante il riconoscimento di attività. Per ultimo, è stato implementato un metodo che gestisce anche quelle attività che il sistema non è in grado di riconoscere, richiedendo direttamente l'intervento del residente. La fase di test e di ottenimento dei risultati è stata effettuata su due dataset di riferimento, ARAS e Kasteren.

Abstract

With the continuous growth of the average age of the world population, it is our duty to constantly research an effective solution to meet the future and unavoidable demands for helping and intervention. The main challenge is to stretch as long as possible the time that an elderly or a fragile person spends living independently in his/her own homes, before making a hospitalization request. This is a sensible theme among the researchers all over the world and, during the years, they proposed many different approaches to solve the problem. More and more technological sciences like Pervasive System or Internet of Things (IoT) allow the development of reliable Ambient Assisted Living Systems (AAL) mounted inside Smart Homes, taking advantage of Home Automation (HA) to provide Assistive Technologies (AT). In this thesis project we present an Ontological Reasoning approach that is able to give support to unsupervised methods present in the literature of Activity of Daily Living Recognition. The core steps of the project have been the analysis, the design and the practical application of an ontology, a Semantic vocabulary, in which we define classes, subclasses, individuals and semantic relations that compose our personal Activity Recognition domain. Finally, we propose a method that manages also the activities that the system is not able to recognize, involving directly the resident. Experiments and results have been tested and obtained on two different datasets of reference, ARAS and Kasteren.

Ringraziamenti

Desidero innanzitutto ringraziare la Professoressa Sara Comai e Andrea Masciadri per il supporto, la disponibilità e l'aiuto che mi hanno fornito durante il lavoro. Un grazie anche a tutto l'ATG Group con cui ho passato gli ultimi mesi.

Un ringraziamento particolare alla mia famiglia, che mi ha permesso di completare questo percorso di studi e che mi ha sempre sostenuto durante l'intero tragitto. Un in bocca al lupo a mia sorella, con la certezza che saprà portare a termine il suo percorso egregiamente, come ha fatto fino ad ora.

Ai miei compagni di Università, con i quali ho condiviso molte gioie per il superamento di esami preparati insieme e molto dolore per le bocciature ricevute. Grazie a Fabrizio, Ilaria e Manan. Un grazie particolare a Mattia, che da semplice compagno del primo anno, si è rivelato un ottimo amico.

Al gruppo di amici, super affollato e pieno di persone fantastiche! Grazie ad Alessio, Andrea, Fabio, Martina, Francesca, Gaia, Gianluca, Giulia, Luca, Sofia, Martina, Luca, Martino, Nicholas, Paolo, Luca, Giada, Cecilia, Mattia, Miriam, Luigi, Cesare e Stefania.

Ad Andrea e Francesco, gli amici di una vita, quelli che sono sempre al tuo fianco e che sai per certo che lo saranno ancora per molti anni, grazie! Un ringraziamento particolare anche alle rispettive donzelle, Paola ed Ilaria, sulle quali so che posso contare nello stesso e identico modo.

Infine, un enorme grazie alla mia ragazza Marta. Grazie per averci creduto fin da subito, per essere entrata nella mia vita in maniera così spontanea e naturale e per avermi accompagnato durante questi due anni STUPENDI! Grazie anche ad Antonio, Elena e Sara per avermi accolto nella lora famiglia.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Monitoring daily activities of elderly people is a major public health problem. According to a US research [1], the world population that is over 65 years of age will increase by 101% between 2000 and 2030 (2.3% each year); conversely, during the same period, the number of family members that can constantly monitor and provide them support will increase by only 25% (0.8% each year). This phenomenon is called "super aging society" or "aging of the elderly population" which means that not only will there be more elderly people, but also that there will be a substantial increase in the number of elderly of an older age. For example, it is estimated that in Italy there will be more than one million people over 90 in the year 2024 and, in China, it is estimated that 330 million people will be older than 65 and 100 million will be older than 80 in 2050 [2]. Therefore, the risk that an elderly lives alone and far away from his/her family will be always greater. In addition, the presence of illness or disability makes each simple everyday action difficult and potentially dangerous. The researchers from all over the world are continuously moving in this direction with the final objective of developing an *Ambient Assisted and Monitored Living system* that allows the elderly to live independently and safety in his/her own home, managing all that services that could recognize his/her general activities and can inform about emergency situations. Fortunately, the continuous growth of sciences as *Pervasive System* and *Internet of Things (IoT)* embraces the idea of smart environment [3], enabling the collection of a large amount and diversified data from an *Home Automation (HA)*. The main focus of this thesis is precisely that of studying the **Activities of Daily Living (ADLs)** in order to detect what happens inside the house to be helpful in the analysis of possible anomalies or incorrect habits (*Behavioral Drift Detection*).

1.1 BRIDGe Project

The Behavioral dRift compensation for autonomous and InDependent livinG (BRIDGe) [4] is a project currently under development at the Assistive Technology Group (ATG) [5] in Politecnico di Milano. The aim of this project is to allow people not completely autonomous to live independently at their homes with their social environment (e.g., family, caregivers, third sector organizations, proximity network), knowing someone they trust is watching over them. Indeed elderly, physically impaired or mild cognitive people are the target of this system to give them all the needed support without any invasive intervention. Different thesis works are to be considered in the frame of such a project, in particular Masciadri et Al.[6] has been of inspiration for the development of this one.

1.1.1 Unsupervised Methods for Activities of Daily Living Drift Modeling and Recognition

Masciadri et Al.[6] work has to be considered the basis for this thesis. Basing on BRIDGe project, they studied and researched an unsupervised model that is able not only to recognize ADLs but also to take over behavioral anomalies. The starting point has been the development of a *Dataset Collector* that converts data published in different formats, in order to provide a standard interface to access them. Since the available dataset are not long enough to caught behavioral changes, the *Dataset Collector* has been extended with SHARON, the Simulator of Human Activities, ROutines and Needs (SHARON) [7], which is a tool developed in the frame of BRIDGe project in order to face the problem of lack of data for *Activity Recognition*. Due to some difficulties encountered in the model of the home automation data generator, a stochastic model that optimizes the collection of an adequate amount of consistent data without wasting too much time and money has been developed. Finally, after a clustering phase, the first results for unsupervised Activity Recognition have been detected: monitoring these results on a long-term period allows the identification of possible Behavioral Drift of the resident inside the Smart Home. In the next sections we will explain how this work has been useful for the realization of this thesis, focusing in particular on the first phase regarding the development of the Dataset Collector and its interface.

1.2 Problem formulation

With the coming of always more powerful technologies it is our duty to give all the possible support to the medical environment: Health Care systems supporting fragile people directly at home is a reality that over the years is more and more required. By monitoring constantly the routine of a person in an apartment it is possible to analyze his/her habits and so to identify if there is something wrong in his/her actions that could be dangerous for his/her safety. Taking advantages of pervasive system as an Home Automation environment, it is possible both to collect useful data and to give all the necessary support and services in case of emergency or need. **Activity Recognition** is the core word of the project and it is still an open challenge for the researchers all over the world: generally, to obtain satisfying results it is necessary to have a large amount of labeled data that has a substantial cost in term of time and money. The research of an optimal unsupervised approach aims to reduce these wastes and to make the model accessible by everyone.

1.3 Thesis contribution

Activity Recognition is a challenge still open among the researchers all over the world: during the years, many and different approaches have been proposed. For this reason, the first step for the development of this project was to understand how the available material and previous works could be improved in an innovative way. The advent of *Machine Learning* has been really helpful for the Activity Recognition field: ascertain performances and limits of supervised methods, nowadays researchers are focusing on unsupervised approaches that allow to optimize the results, avoiding waste of time and moneys. My personal contribution with this thesis project is the development of a **Semantic Layer** that could be integrated in existing models to give them support for recognizing the correct activity carried out by a person inside a *Smart Home*. Taking inspiration from the most common datasets made available from other researchers, a semantic vocabulary (**Ontology**), has been built. Here all the elements, individuals and relations that concern the Activity Recognition field are resumed, creating a common domain of interest. Then, Masciadri et Al.[6] work has been configured to support the integration of this layer: querying the ontology with a sequence of active sensors, it retrieves the most probable activity carried out by the resident. Indeed, the main advantage in using an ontology is precisely that of being able to be adapted, extended or integrated according to need. Finally, two simulations on different datasets provide results on the potentiality of the proposed approach.

1.4 Definitions

Here we provide some useful definitions to better understand future concepts that we will meet during the reading of this thesis.

Semantics

We will encounter this term in different contexts: *semantic domain, semantic web and semantic relations*. But what is semantics? Semantics is the linguistic and philosophical study of *meaning*. In languages, programming languages, formal logics and semiotics it analyzes and interprets symbols and signs to create a common domain of interest. This domain, from the point of view of computer science and semantic web, is called **ontology**. Here, it is possible to find all the information, features and relations that belong to the individuals that populate it.

Resident

People that live inside the smart home are often cited simply as residents. It is important to specify that the whole project is thought and designed to reach the final target of being helpful for the category of people that needs some external support to live safely and independently inside their houses. This “fragile people” category includes elderly, mild cognitive or physically impaired people and the ones with mild chronic diseases.

Behavioral Drift

Each person, during his/her daytime, carries out a great number of simple and complex actions. The combination of this actions gives as result the generation of a Daily Activity. By monitoring the resident inside the house on a long-term period it is possible to build his/her personal pattern on his/her everyday routine. A change on how the resident carries out a particular activity or on his/her psychophysical status during its fulfilment may be an indicator of decay or illness that the system is able to report and to issue a request of assistance.

Sensorset

Data acquired by the sensors inside the smart home consist in a sequence of sensors activations over time. For the scope of this work we consider only boolean sensors, where the status are intended as *"activated"* or *"not activated"*. A sensorset could be considered as a snapshot of the status of the sensors inside the house at a given timestamp. It could be represented by a boolean vector of N elements, where N is the number of the sensors mounted in the house. For every second t, its corresponding sensorset ss_t takes the form of:

$$ss_t = \{s_1,\ s_2,\ ...,\ si\}\ \epsilon\ \{0, 1\}$$

where s_i is the status of the i-th sensor.

1.5 Thesis organization

Next chapters are organized as follows:

- In Chapter 2, we introduce all the previous and useful works concerning the topic of this thesis project. We explore in details the typology and complexity level of the nowadays Smart Homes, which sensors populate them and how they are configured for Activity Recognition; supervised approaches dominate again in this field, but some unsupervised proposals are interesting and they deserve to be taken into account: precisely an unsupervised method is the basis for this work.

- In Chapter 3, we explore the Semantic Web world. Starting from its history, we explain what is it, for which reasons it was born and which are its principal potentialities; here, we find also how the Semantic Web can affect our Activity Recognition domain.

- In Chapter 4, the proposed model is presented. The ontology design and development have been the first and the most important steps; a filtering phase anticipates the simulation process that gives us the final results on Activity Recognition.

- Finally, in Chapter 5 we re-analyze the entire work explaining which are its limits and how it could be improved with future features.

Chapter 2

State of the Art

In Chapter 1 we faced the "super aging society" problem and we explained how much the research in the Assistive Technology field is important for our future: it is our duty to support fragile people in their everyday simple actions, allowing them to live independently and safety directly at home and preventing, when possible, the development of diseases. Instead, in the first section of this Chapter we will focus on the concept of *Smart Home*: what is it? How many different typologies of smart homes exist? Then we will present the sensors that populate these environments, explaining for which reasons we have chosen and focused only on some categories considered more suitable for an *Health Care System*, discarding the others. Subsequently, we will discuss about related projects and methods encountered in the literature for *Activity Recognition*. Finally, we will show how the datasets made available from the researchers all over the world are organized to be helpful for the growth of the Activity Recognition domain.

2.1 Smart Home

Over the course of the 20th century, individuals spend always most time in their homes or workplaces; indeed nowadays, inside these environments, there are all the technologies and comforts that allow the residents to carry out their daily routine without too many efforts or the need of going out. The term *Smart Home* refers to an environment in which an **Ambient Intelligence** system is configured to manage all the home automation devices present in it. "*What makes a home Smart are the interactive technologies that it contains*" [8]. The continuous growth of sciences as

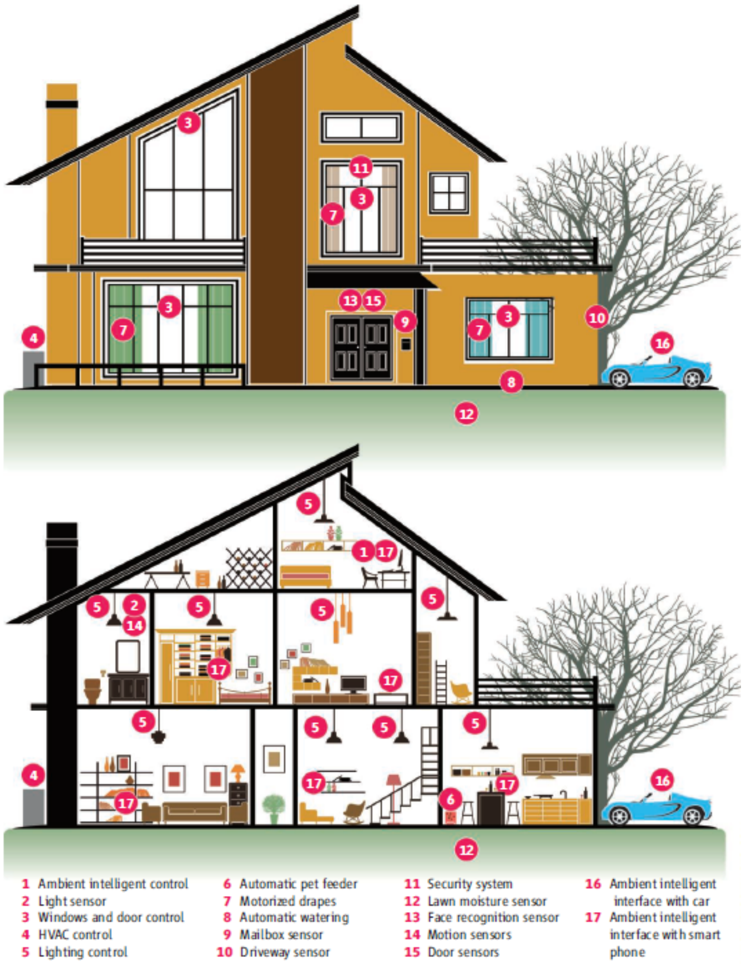


Figure 2.1: A Smart Home example, with the disposition of all the automation devices mounted inside it.

Pervasive Systems or **Internet of Things** is bringing such Ambient Intelligence systems in the home closer to reality insomuch to be invisible for the residents that live inside it [9]. As a matter of fact, the continuous miniaturization of microprocessors allows the manufacturers to mount them inside common objects of daily use; moreover, a great diversification in the costs of the devices exists, making a Smart Home available not only for the rich class but for every level of society. Figure 2.1 shows an example of a Smart Home equipped by an Ambient Intelligence system: lighting, windows and doors control are just some of the supported actions. Furthermore, thanks to the advent of increasingly powerful smartphones and mobile devices it is possible to communicate with such systems remotely: for example, through an application

Figure 2.2: Five different Smart Homes applications.

on the smartphone it is possible to inform the Home that the resident will come back from work at 19.00 and that it has to start the heating system one hour early. At the beginning, the Artificial Intelligence(AI) of the system had to communicate constantly and directly with the resident: for example, he/she manually has to set up the intensity of the light inside a room in a particular hour or phase of the day. However, after a while, the system learns itself which are the habits and the preferences of its inhabitant and it carries out some decisions or ambient changes independently. At the end, the interaction between the AI and its resident aims to be as minimum as possible, making the home not too much intrusive in his/her daily routine. During the years, this aspect has opened new scenarios on which a Smart Home could be applied and, nowadays, five main categories (Figure 2.2) have been identified:

- *Security and Control*;
- *Appliances*;
- *Entertainment and Connectivity*;
- *Energy and lighting*;
- *Health monitoring*.

The differences between these five application domains is not always well-defined and sometimes their features overlap each other. However, for the scope of this project, our focus is only on the Health monitoring scenario and how it can transform a Smart Home in an **Health Care** system useful for *fragile* residents.

2.1.1 Health Care Systems

Sensors and automation devices mounted in an ambient intelligent home could provide useful information about the health status of its dwellers. For example, thanks to motion sensors scattered through the environment, researchers analyze parameter

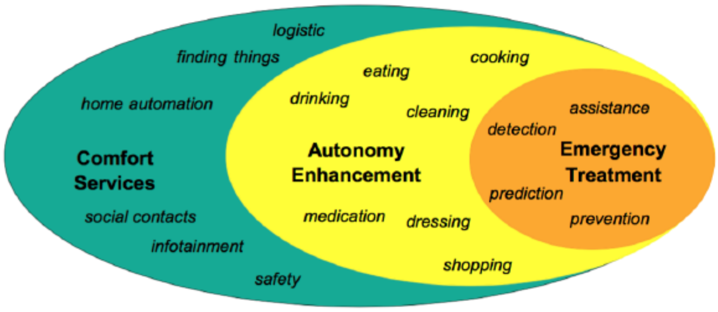


Figure 2.3: Three Health Care System categories.

such as walking speed to identify, over multiple year, changes in mobility pattern that could be a first symptom of dementia; other researchers exploit the ambient intelligent system to identify early-childhood screening for autism. Becker et Al. [10] divide the structure of an Health Care System into three further categories shown in Figure 2.3:

- *Emergency Treatment* can be considered the kernel of any Health Care System; its services constantly monitor the situation inside the home with the aim of predicting, recovering or alerting propagation of emergencies;
- *Autonomy Enhancement* automates all that daily actions in which the resident could meet some difficulties in their realization; this class enables to live independently inside the home without the intervention of medical or social care personnel;
- *Comfort Services* are all that services that are not included in the previous categories: their aim is to enhance the quality of live of the dweller.

2.2 Automation Devices

The continuous growth of sciences as Pervasive System or Internet of Things allows the creation of networks of interconnected sensors always more complex and, at the same time, more efficient. Nowadays, the dimensions of embedded microprocessors are insomuch small that they can easily scattered through the environment going unnoticed or they can be mounted on common object of daily use; moreover, to communicate each other, they do not need particular infrastructures or wires but they can be included in the home creating a **Wireless Home Automation Network**

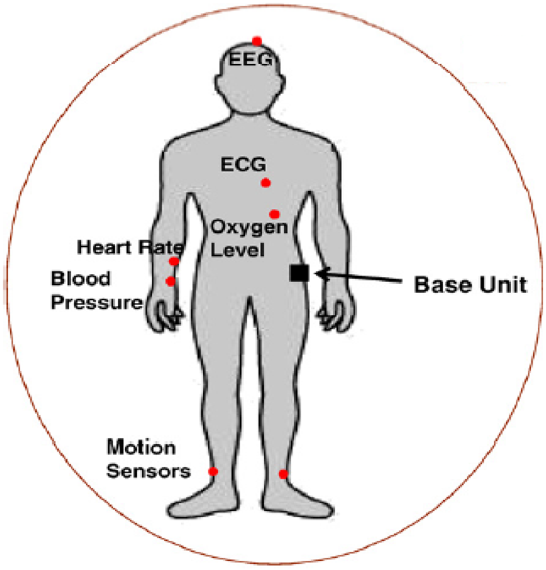


Figure 2.4: Body sensors example.

(WHAN) [11]. At the moment, the number and the typology of automation devices that populate a Smart Home, more easily identified as sensors, is really large and diversified depending on prices, dimensions and utilization. We can group up all these types into two main categories:

- *Body Sensor networks (BSN)*;
- *Environment Sensor networks (ESN)*.

2.2.1 Body Sensors

The Body Sensors category refers to all that sensors that can be included in wearable objects or that can be applied directly on the person, for example, under the clothes. From a medical point of view they are really useful because they permit to avoid emergency situation such as falls or hazards and to monitor constantly the patient during therapy or rehabilitation processes. Inside a watch, for example, can be mounted a sensor that measures the heartbeat of the patient. Figure 2.4 shows some of the most common sensors used [12]: ECG sensor, EEG sensor, EMG sensor, accelerometer/gyroscope, CO2 gas sensor, blood glucose and pressure, pulse oximetry, humidity and temperature sensor. Unfortunately, for the scope of this project, Body sensors category has some limits. The main disadvantage in using this typology of sensors is that the resident must keep in mind to wear them: this is not an obvious think because, for example, Alzheimer patients may forget to use them. Furthermore, our analysis is on a long-time period and this means that the patient has to continuously change the batteries, since they have not a fixed power supply. Finally, they are certainly intrusive and they may affect the actions of the patient. For these reasons we have decided to focus mainly on *Environment Sensors* category.

2.2.2 Environment Sensors

The Environment Sensors category is more recommended for Behavioral Drift detection and Assisted Living, since the patient is not required to perform particular tasks or actions for their correct working. Indeed, this category refers to all that sensors scattered in the environment and with a more or less fixed position: the resident may even be not informed about sensors around him/her. The most used environment sensors are:

- *Infrared Motion and Ultrasonic* used to notice movement of people or objects inside a room;
- *Sonar* used to detect the presence of a person in a room;
- *Camera and Microphone* used to observe remotely the situation inside the home and to record audio and videos;
- *Force and Pressure* are positioned under chairs, couches ore bed to identify if someone is over them;
- *Magnetic switches* to control doors movement;
- *Contact* are used to detect if doors or cupboards are closed or open;
- *Proximity* identify if someone or something is closed;
- *Radio Frequency IDentification (RFID)* used to obtain information from object marked with a tag ;
- *Smoke, Gas, Temperature, Humidity and Luminance* to collect information about the environment.

In this project, we have not used cameras and microphones since they violate the resident privacy and, as for body sensors, because they have been considered too much intrusive.

2.3 Activity recognition approaches

Veronese et Al.[3] define Activity of Daily Living (ADL) as "*the actions of the everyday life, what people normally do in daily living including any activity, such as feeding themselves, bathing, dressing, grooming, working, doing housework and leisure*". Recognizing an activity carried out by a person through a smart environment based on data collected by sensors is not a simple task. Indeed, some common issues emerged from the literature in Activity Recognition field could be resumed in the following four definitions:

- **Concurred activities**: people sometimes are used to perform more than one activity at the same time such as talking on the phone while they are cleaning the bathroom;
- **Interleaved activities**: it is really common that an individual stops what he/she is doing in that moment to perform other tasks such as answering the phone or opening the door returning later to complete the interrupted activity;
- **Ambiguity of interpretation**: some actions are common to different activities, making the recognition process difficult: if a person is lying on the couch he/she could read a book or have a break or sleep and so on;
- **Multiple residents**: if in the home are present more than one person the system has to distinguish which are the sensors activated by one resident and which are the ones activated by the second one.

Many approaches to perform Activity Recognition have been proposed in the literature; Machine Learning is widely diffused in this field and two main research categories have been identified: Supervised and Unsupervised approaches.

2.3.1 Supervised approach

Supervised learning is a Machine Learning technique that aims to educate a system in order to allow it to solve tasks autonomously on the basis of a series of input variables

X and an output variable Y, finding and creating a mapping between them. They are still the most diffuse approaches chosen by the researchers all over the world due to the fact that they are able to reach acceptable and high results. However, the main disadvantage in using supervised methods is that they require, to work well, the availability of a great number of annotated training data. This labeling phase is a complex task that needs, how we will explain in the next section, a lot of time and money to be produced. Supervised methods can be divided into three different classes:

- **Generative methods** are based on probabilistic algorithms and graphs that manage both the generation of activities data and their labeling process. Among them the best known are *Naive Bayes Classifier* (NBC), Hidden Markov Model (HMM), dynamic Bayes networks and Gaussian Mixtures [13];

- **Discriminative methods** work on the boundary between different labeled classes. We met Decision Tree (DT), Support Vecotr Machines (SVM) and Conditional Random Fields (CRF) algorithms [14] [15];

- **Clustering methods** work on the distance computed between sequences of sensor events (e.g., k-NN classifier) [13].

Cook [16] presents a project in which tests 3-fold cross validation (NBC, HMM and CRF) over the set of annotated activities acquired from 11 different datasets of CASAS project [17]; the averaged results are: 74,87% for NBC, 75,05% for HMM and 72,16% for CRF. However, Cook shows also that, in addition to the labeling phase, another important disadvantage in these supervised approaches is that the results, testing the same models on apartments with multiple residents, are much lower than the previous one.

2.3.2 Unsupervised approach

Unsupervised learning is a Machine Learning technique that consists in providing the system with a series of inputs (observations X, system experience) that he will reclassify and organize on the basis of common characteristics to try to make reasoning and classification on subsequent inputs. The main difference between the supervised approach is the one that make these methods so difficult and still in development: the absence of labeled data. Indeed, the system has to automatically learn to which class an input element belongs. At the moment, in the literature, we have identified different novel proposals:

- **Computer vision**: these methods acquire data through cameras and proposed recognition algorithms of web images [18];
- **Smartphone accelerometer**: these method acquire data taking advantage of technologies already present in daily common object like smartphones [19];
- **Clustering methods**: for example, Bannach et Al. [20] use clustering techniques to add a new sensor in an already existing network and the system is able to adapt itself and to recognize the new entity in an unsupervised manner;
- **Knowledge-Based** (KB): till now we met only *data-driven* approaches, or rather that approaches that take advantage of measured and acquired data; instead, knowledge based methods exploit relations and individuals present in a semantic domain [21] [22] [23].

In particular, this last category has been of inspiration for the development of this thesis project. Riboni et Al. [24] [25] implement an innovative system (Knowledge-based Collaborative Active Learning for Activity Recognition (NECTAR)) in which they built a personal *ontology* to give support to a *Markov Logic Network* model in the activity recognition process. Moreover, they improves their work also with an *Activity Learning* entity which queries the resident when the system is in doubt with the final decision. We will see in details the concepts of **Ontology**, **Semantics** and Knowledge-based in Chapter 3.

2.4 Real datasets collection

The "super aging society" is a theme always more sensible for the researchers all over the world. During their projects and methods on Activity Recognition field, they provide, over the years, many and different datasets that allow them to test in the reality their works. These data are usually acquired by the streaming of sensor activations in the reference environment and, when possible, with the addition of the ground truth of the activity carried out by the resident. These datasets, also if they are designed for different apartments and projects, present some common concepts that worth to be mentioned:

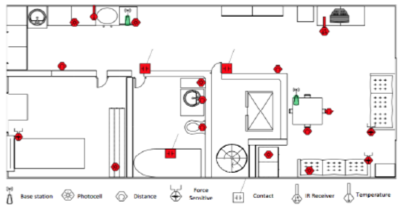
- **Data collection:** in this phase the sensors status inside the home (active or not active) on a defined period of time is acquired. To collect these data some projects recreate a *Real Living Scenario* (RLS) thanks to the collaboration of fragile people that accept to live inside the home for some days; instead, other projects organized *Designed Experiments* (DE) in which many participants are asked to carry out specific actions or activities: for example, it is required to ten different individuals to make coffee in the same room, but in different moment, taking memory of the pattern of actions that a single person has followed to reach the target.

- **Sensor types:** sensors are not divided only for their type categories, but also for the way in which they acquire data. Some sensors transform the acquired data into a voltage that has to be traduced by the system interface; indeed, other sensors trigger only their status: 1 if it is active, 0 if it is not. Sensors of this last category are called *Boolean Sensors*. For the scope of this project we mainly focused on this typology, paying attention to the difference between "*sensor status*" and "*sensor event*". The first refers to the exact value of its variable in a given instant, while the second identifies the sensor as active for all the period in which its monitored variable does not change its status. For example, when we turn on or off a light we invert the status of its sensor when we press the switch; instead, when we are sitting on the couch its sensor will indicate 1 until we will not stand up.

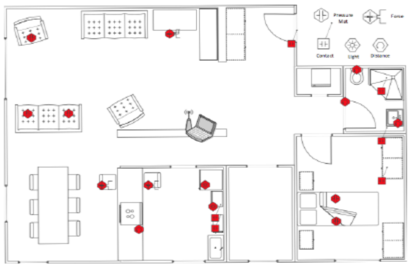
- **Number of residents:** some experiments may consider not only one person inside the home, but multiple residents at the same time.

- **Ground truth:** the labeling phase of raw data on the ongoing Activity is a crucial step. To solve this difficult task different approaches have been proposed in the literature that we can group up in three different categories: in the first one the resident is required to take note of its ongoing activity through a specific GUI positioned inside the home; other projects monitor the situation from outside thanks to videos and audios recorded by camera and microphones, manually labeling the dataset; finally, automated methods infering the activities from data.

Following some of the most popular datasets used for Activity Recognition:



(a) House A



(b) House B

Figure 2.5: Two houses of ARAS dataset.

Figure 2.6: House C of Kasteren dataset.

ARAS (Activity Recognition with Ambient Sensing) is a dataset collected is a human activity recognition dataset that is collected from two real houses (Figure 2.5) with multiple residents, during a period of two months. It contains the ground truth labels for twenty seven different activities and each house is equipped with twenty environmental binary sensors of different type.

KASTEREN Tim Van Kasteren works on several datasets with different features, starting from the duration of data acquisition to finish with a great variance of sensors type. Since we are interested mainly on *Ambient Assisted Living* we have take in consideration his project on an apartment (*"House C"*) disposed on two floors and with only one resident (Figure 2.6).

CASAS (Center for Advanced Studies in Adaptive Systems) is a department of the Washington State University very famous and operative in the Activity Recognition field. Diane J. Cook is one of the main exponents of this department and her works have been points of reference also for this thesis project. Starting from the Smart Home in a Box (Shib) project [17], their final aim is to design a system "small in form, lightweight in infrastructure, extendable with minimal effort, and ready to perform key capabilities out of the box". A lot of different apartments, combination between their residents and also considering the presence of pets have been tested over the years and all these datasets are free published on the official web site [26].

DOMUS is a dataset entirely collected through DE approach [27]. Several individuals have been asked to carry out specific actions on a period of 11.5 hours. Respect previous dataset, here the number of sensors present in the environment was very high (up to 78 different sensors).

MIT is a dataset composed by two houses (Figure 2.7) and a single resident monitored for two weeks [28]. Also here the number of sensors is very high (up to 85) and they are all boolean sensors. The labeling method is different from other approach: a PDA is given to the user who is queried every 15 minutes about the ongoing activity.

In this project we have used and tested our model on ARAS and KASTEREN datasets. We will discuss more in details about them more in Chapter 4.

(a) House 1 (b) House 2

Figure 2.7: Two houses of MIT dataset.

Chapter 3

Semantic Web

"Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully"[29]. This is due to the fact that, nowadays, computers are able to parse Web pages for simple actions like layout and routine processing but in general, they are not able to process the *semantics*. The *Semantic Web* aims to share a common environment in which programs and software agents can browse from page to page for extracting all that information useful to satisfy a complex user's task. *"Such an agent coming to the clinic's Web page will know not just that the page has keywords such as "treatment, medicine, physical, therapy" but also that Dr. Hartman works at this clinic on Mondays, Wednesdays and Fridays and that the script takes a date range in yyyy-mm-dd format and returns appointment times."* This semantic relations must not be a separated version of the Web, but rather an extension of the current one, in which information is given in well-defined meaning, permitting people and computers to work together. So far, the Web has grown most rapidly as an intermediary layer between documents and people rather than for data and information that can be processed automatically: the Semantic Web aims to bridge this gap. The *World Wide Web Consortium (W3C)* totally embraces this mission. It is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards [30] based on five principles:

- *Web for All*: makes the human communication, commerce and opportunities to share knowledge accessible by everyone, breaking all the possible hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability barriers [31];

- *Web on Everything*: the number of different kinds of devices having access to the Web has grown vertiginously and it has to be accessible anywhere, anytime, using any device [32];
- *Web for Rich Interaction*: the Web is born as a communication tool and for many years it has been used with a "read-only" function; blogs and wikis brought more authors to the Web, and social networking emerged from the market for content and personalized Web experiences;
- *Web of Data and Services*: the Web is both a giant repository of linked data and a set of services that exchange messages; these two visions work in a complementary way depending on the applications and need;
- *Web of Trust*: our communication ways are totally changed in the last years; through the Web we carry out commercial and personal relationship, often without meeting in person. We "meet on the Web" and, for this reason, security and privacy are two key aspects that have to be defended.

3.1 Inference and Semantic Reasoner

To better understand how the Semantic Web works it is fundamental to introduce two basic concepts: what is an **Inference** and what it is used for? On the Semantic Web, data is modeled as a set of *relationships* between resources; inference means that automatic procedures that allow to discover and to generate new relationships based on data and based on a set of additional information as e.g. a rules set. These additional information can be defined through a *Vocabulary*, a collection of concepts and relationships used to describe a particular area of interest, and can be elaborated by a *Semantic Reasoner*, or simply a **Reasoner**, which is a software that plays the role of inference engine, able to infer logical consequences from the set of asserted facts or axioms.

3.2 Knowledge Representation

Structured collections of information and sets of inference rules useful to compute automated reasoning are the key aspects that allow the Semantic Web to function. During the years, before the advent of the Web, such systems have been studied by artificial-intelligence researchers all over the world and they call this technology *Knowledge Representation*. At the beginning, traditional knowledge-representation systems require people to share the same definition of common ideas and concepts, making the system as centralized as possible; however, after a while, this centralization becomes out of control due to the fact that the size and the scope of such a system rapidly becomes unmanageable. Another limit was that these systems voluntarily and carefully filter the questions that can be asked, making the computer able to answer reliably, without risking to fall in *paradox*. Indeed, recalling Gödel's theorem from mathematics: *"any system that is complex enough to be useful also encompasses unanswerable questions, much like sophisticated version of the basic paradox"*. To prevent this problem, traditional knowledge-representation systems improve their own personal set of rules for making inferences about their data, limiting on the other side the communication between them: the data could be transferred from one system to another, but the rules, having a completely different configuration, usually could not. On the other side, Semantic Web researchers consider paradoxes and unanswerable questions the price that must be paid to reach versatility and they accept the challenge of providing a language that expresses both data and rules for reasoning about the data and that allows rules from any existing knowledge-representation system to be exported on the Web. At the moment, the task of the semantic web community is that of integrating the Web with the **logic**: it must be powerful enough to describe complex properties of objects, but it has to pay attention to not easily fall in paradoxes. Considering the previous features, two important technologies are available for developing the Semantic Web: *eXtensible Markup Language* (XML) and the *Resource Description Framework* (RDF) [33].

3.2.1 (XML) eXtensible Markup Language

This language is already widely known in the Internet community and is employed in a great number of software development activities. Differently from HTML, which was designed for hypertext documents with fixed structures, XML focus its eyes on documents of arbitrary structure. A well-formed structure of a XML document is composed by a balanced tree of nested sets of open and close tags, and each of this tag can include different attribute-value pairs. The main advantage is that there is not a predefined or fixed tag vocabulary and combinations, allowing everyone to personalize them according to need. Figure 3.1 shows an example serialization of part

```
<class-def>
    <class name="plant"/>
    <subclass-of>
        <NOT><class name="animal"/></NOT>
    </subclass-of>
</class-def>
<class-def>
<class name="tree"/>
    <subclass-of>
        <class name="plant"/>
     </subclass-of>
</class-def>
<class-def>
      <class name="branch"/>
      <slot-constraint>
            <slot name="is-part-of"/>
            <has-value>
               <class name="tree"/>
            </has-value>
      </slot-constraint>
</class def>
```

Figure 3.1: Partial XML serialization of a vocabulary example

of a semantic vocabulary for the content *”plant”*. In the labeled tree representing the basic XML data model each tag corresponds to a labeled node in the model, and each nested subtag is a child in the tree. This example shows only one possible XML-based syntax for the vocabulary; however, XML is used principally for defining grammars and due to the fact that different grammars can be used to describe the same concept, XML allows multiple serializations. Figure 3.2 shows a totally different form of the same definition of the one in figure 3.1.

```
<class-def>
        <name>branch</name>
        <slot-constraint>
                <name>is-part-of</name>
                <has-value>tree</has-value>
        </slot-constraint>
</class-def>
```

Figure 3.2: Different XML serialization of the same vocabulary example

3.2.2 Resource Description Framework (RDF)

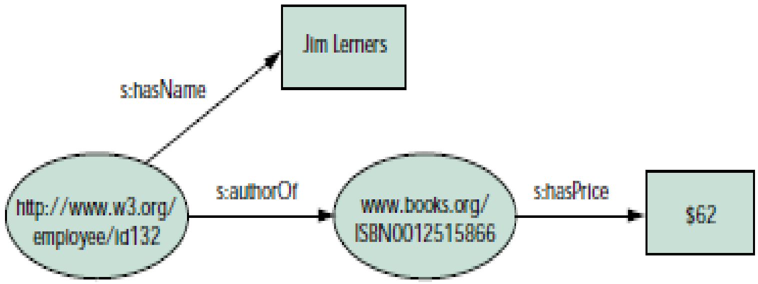


Figure 3.3: RDF Graph

RDF is a standard model for data interchange on the Web. It has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed [34]. The basic building block in RDF is a triple of the type object-attribute-value $A(O, V)$, that means that an object O has an attribute A with a value V. Another standpoint to see this relationship is as a labeled edge between two nodes: $[O] \rightarrow A \rightarrow [V]$, where both the relationship between things as well as the two ends of the links are identified by URIs. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This **RDF graph** view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations. In figure 3.3, for example, the graph expresses the following three relationships in $A(O, V)$ format:

hasName ('http://www.w3.org/employee/id1321', "Jim Lerners")
authorOf ('http://www.w3.org/employee/id1321',
'http://www.books.org/ISBN0012515866')
hasPrice ('http://www.books.org/ISBN0012515866', "$62").

As happens in XML, also **RDF schema** defines a particular vocabulary for RDF data (e.g., *authorOf*) and specify the kinds of object to which these attributes belong to. For example, it is possible to define that the object *"ISBN0012515866"* is of the type *rdf:type book* by creating a type *arc* referring to the *book* definition in the RDF schema:

<rdf:Description about="www.books.org/ISBN0012515866">
<rdf:type resource="http://description.org/schema/book">
</rdf:Description>

This RDF schema mechanism provides a basic type system for RDF models: it uses some predefined terms, such as *Class, subPropertyOf, Property* and *subClassOf* that will be all introduced in the next paragraph about Ontologies.

3.3 Ontologies

Till now, the term *vocabulary* has been used without giving a well-defined meaning: on the Semantic Web, vocabularies define concepts and relationships, known as *"terms"*, used to described and represent an area of concern [35]. Inside a vocabulary, the terms used in a particular application are classified, new relationships are discovered and constraints on their usage are defined. Depending on the needs, a vocabulary can be very complex, composed of thousand of terms, or less intricate, defining just few concepts only. It's right here that it is possible to meet the thin difference between a **Vocabulary** and an **Ontology**: the term *"ontology"* is generally adopted to identify a more complex and quite formal collection of terms, while a *"vocabulary"* is preferred when the previous formalism is not needed. Gruber[36] views an ontology as a *conceptualization* on which a body of formally represented knowledge is based: *"a conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose"*, grouping together all the objects, concepts, and other entities with the relationships that hold among them. But why would someone want to develop an ontology? One of the stronger goal in developing ontologies is *to **share** common understanding about a reference domain helping data integration between different datasets*. Suppose, for example, the application of the ontologies in the health care field. At one side, doctors and medical professionals use them to represent knowledge about symptoms, diseases and treatments; on the other side, pharmaceutical companies use them to represent information about drugs, dosages and allergies. If the relative web sites make public the same underlying ontology of the terms they use, the two knowledge could be merged to allow people or software agents to extract from them useful and aggregate information to answer user queries or as input data to other applications. Figure 3.4 shows a possible result of an ontology in the health care field [37].

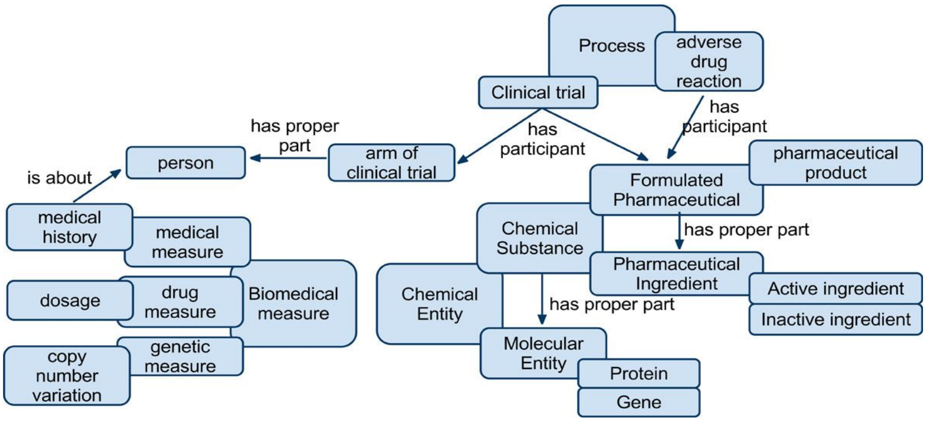


Figure 3.4: An example of a medical ontology.

Starting from this basic feature, it is possible to highlight five other main advantages among the several that the development of an ontology gives us:

- Everyday, environment like e.g., libraries, museum or newspaper have to manage a large amount of data; ontologies **organize** this knowledge simplifying this task and leveraging the power of linked data;
- Once a common domain knowledge has been developed, it could be **reused** from other researchers that need it; furthermore, it is possible to integrate several existing ontologies describing portions of a unique domain;
- Making **explicit** domain assumptions allows us to easily change these assumptions if the knowledge about the domain changes;
- **Separating** the domain knowledge from the operational knowledge; "we can describe a task of configuring a product from its component according to a required specification and implement a program that does this configuration independent of the products and components themselves" [38];
- The domain knowledge can be **analyzed** once a declarative specifications of the terms is available to value if the ontology is suitable for re-use or for some extension.

3.3.1 Building an Ontology

Once it has been clarified what an ontology is and what are the advantages that its development involves, it is useful to understand which are the first steps that

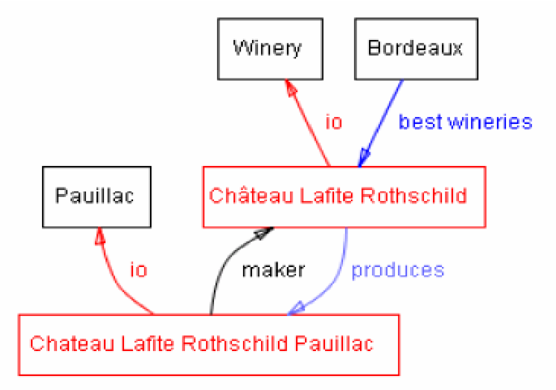


Figure 3.5: An example of the wine domain. The black identify classes while the red identify instances. Direct links represent object properties and internal links instance-of and subclass-of.

must be carried out during the building of an ontology. Following the well-formed *101 guide* [39], an ontology is composed by a formal description of concepts in a domain of interest (**classes**) , by properties of each concept that describe particular features or attributes of the concept itself(**object properties**) and by restrictions on the object properties that define its type(**data properties**). Merging together these three definitions with a set of **individual instances** of classes we obtain the final result of a **knowledge base (KB)**. In this guide, McGuinnes et all. use, for example, an ontology about wines and their perfect pairing with food. The main focus of most ontologies is the identification of classes, the concepts in the domain. In the wine field, for example, we can divide the class of all wines into red, white and rosè wines. Furthermore, a class can have a **subclass** that represents more specific concepts with respect to its **superclass**. Sparkling and non-sparkling wines could be a more detailed view of the red, white or rosè classes; instead, the single glass of Bordeux wine on the table is an **instance** of the class Bordeux wine, subclass of Red wine. About object properties of classes and instances we can identify, for example, some attributes like *flavor*, *body*, *sugar level* or *maker*: *Chateau Lafite Rothschild Pauillac wine* has a full body and it is produced by *Chateau Lafite Rothschild*, an instance of the class *Winery* (Figure 3.5). After this brief description of the main elements that compose an ontology it could seem that the process of an ontology development is the same of the one adopted during an object-oriented design phase. This is not true. In an object-oriented project, the programmer focusses primarily on the methods of the single class making design decisions based on the *operational* properties of that class; indeed, a KB designer makes these decision based on the *structural* properties of the class, taking into account all the possible interactions with present and future elements of the domain. There is no one correct methodology for developing ontologies, but it is possible to detect seven steps along which we can build an ontology that reflects that concepts of the reality of the world for which it has been designed:

1. The first step is to **determine the domain and the scope of the ontology**. Basic questions like *"for what we are going to use the ontology?"* may help in this phase and their answers my change during the design process. This is especially useful because it limits the borders of the model, which otherwise could continuously grows exceeding from the original scope;

2. Consider the **reusing of existing ontologies** is not a shame; use the work of someone else may be a requirement if our project has to interact with other applications that have already their own ontologies or vocabularies (*interoperability*);

3. It is important to have in mind more or less the list of principal terms and properties of the model; in this way we can **enumerate** them, helping us to give a dimension to our problem;

4. **Define classes and class hierarchy** is a crucial point; there are different classic approaches to do this: *top-down*, *bottom-up* or a *combination* between the two processes. The top-down approach starts with the definition of the most common or general concept of the domain and then focusses on the specialization of the others; the bottom-up does exactly the inverse, starting from the most specific classes with subsequent grouping of these into a more general concepts(the leaves of the hierarchy); the combination of the two approach defines first the main concepts and then specializes and generalizes them appropriately; In the Figure 3.6 there is an example of the three approaches in the wine example;

5. Once the class hierarchy has been designed it is necessary to **define the object properties**: classes alone do not provide enough information about concepts. It is important to highlight that all the subclasses of a class inherit its properties and, for this reason, an object property should be assigned to the most general class that can support it;

6. In addition to the object properties, to reach another level of detail, we have to **define also the data properties**; what is the minimum or maximum cardinality of the object? And what is its type? All these attributes can be added to the ontology, stating, for example, that the *grape* property of a Wine has a minimum cardinality of 1 and that the *price* of a wine has the value type *Number*, in particular *Float*;

7. The last crucial step is that of **creating individual instances** for each class. Then, we can connect our classes through the properties defined in the steps 5 and 6, assigned to the property the starting class called **domain** and the end class called **range**.

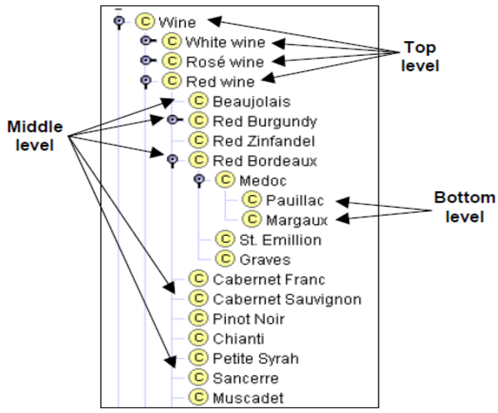


Figure 3.6: The three class hierarchy approach; the combination of the top-down and the bottom-up is identified by the Middle level.

3.3.2 Tools and Libraries

At the moment, several tools for the design of an ontology exist. For the scope of this work I have chosen Protégé(5.0) [40], an open source ontology editor developed by the Stanford University School of Medicine. It is a framework thought for building intelligence systems and it *"is supported by a strong community of academic, government and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce and organizational modeling"*. The knowledge-based representation that has been chosen is the *classic* one, based on a description logic approach, an evolution of the first-order-logic. Once the design phase is terminated, to manage the ontology in our system in Java it has been used the

OWL API [41], which gives us all the means to handle all the basic operation like *load*, *save* and *modify* our ontology.

Chapter 4

Experiment and Results

Semantic Web is a world continuously growing with a really high potential of being used for a lot of applications in many different fields. Once clarified what is it and which are its principal features, we have to think if it is useful to apply the *semantic* to our personal model and what are the advantages that it takes to the work. In this chapter we will go through each phase of the thesis project, describing which is its personal contribution to reach the final target of discovering the correct *Activity of Daily Living*. Figure 4.1 sums up the principal steps carried out during the work: each entity in the schema represents a future section of this chapter .

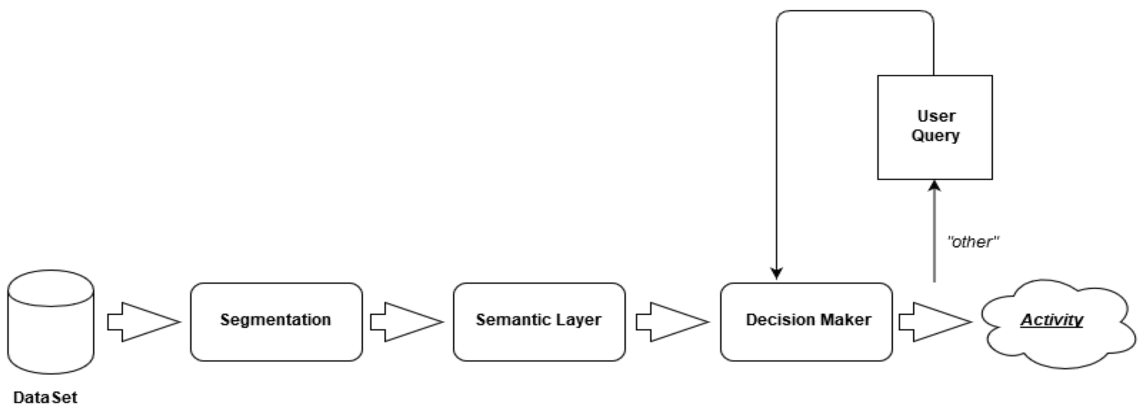


Figure 4.1: Schema of the model developed in this thesis project

4.1 Real Dataset and Activity Segmentation

The first phase of the work has been the analysis of the real datasets used from Masciadri et Al. [6] to compose their *Dataset Collector* entity. Here, each dataset is modified and parsed in a unique format that allows the general model to work well with any different datasets used as input. In this way we have not to worry about the input data type, but in any case they must be individually analyzed to understand, for example, which sensors have been positioned and what is their type, which activities are supported in the smart environment and what is the number of people that live inside the house. Researchers all over the world make many datasets available useful for automatic *Activity Recognition*, but in this project the focus is mainly on two of them: *ARAS Dataset* [42] and *Kasteren Dataset* [43].

4.1.1 (ARAS) Activity Recognition with Ambient Sensing

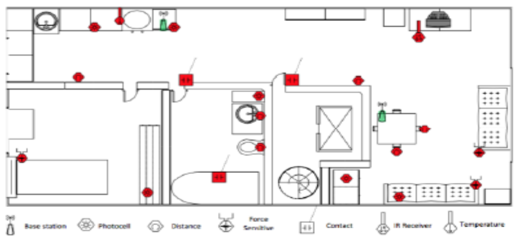
Other	Going Out	Preparing Breakfast	Having Breakfast
Preparing Lunch	Having Lunch	Preparing Dinner	Having Dinner
Washing Dishes	Having Snack	Sleeping	Watching TV
Studying	Having Shower	Toileting	Napping
Using Internet	Reading Book	Laundry	Shaving
Brushing Teeth	Talking on the Phone	Listening to Music	Cleaning
Having Conversation	Having Guest	Changing Clothes	-

Table 4.1: ARAS Activity Supported

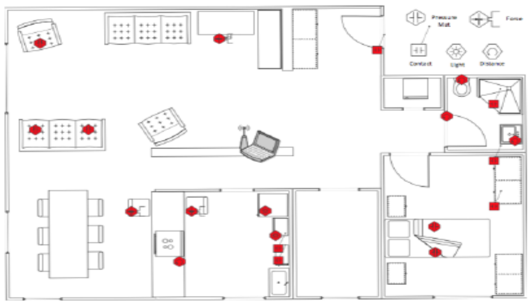
ARAS is a human activity recognition dataset that is collected from two real houses with multiple residents, during a period of two months. It contains the ground truth labels for twenty seven different activities (Table 4.1) and each house is equipped with twenty environmental binary sensors of different type. For the scope of this project we choose the ARAS acquisition format as the basis for the unique interface developed by the Dataset Collector. Figure 4.2 shows the disposition of the sensors inside the house with a legend of their type:

- *Force Sensors* or *Pressure Mat* are positioned under beds and couches identifying if someone is present over them;
- *Photocell Sensors* mounted on drawers, fridge or wardrobes;

- *Contact Sensors* detect if doors and cupboards are closed or open;
- *Proximity Sensors* identify if someone or something is closed to the sensor;
- *Sonar Sensors* detect the presence of a person in the room;
- *Temperature Sensors* mounted on the oven in the kitchen;
- *Infrared Receiver* positioned on the TV.



(a) House A



(b) House B

Figure 4.2: ARAS Houses

4.1.2 Kasteren Dataset

Tim Van Kasteren works on several datasets with different features, starting from the duration of data acquisition to finish with a great variance of sensors type: his focus is not only on environmental ones but also on motion detectors, reed switches, cameras, accelerometers and RFID readers. Since we are interested mainly on *Ambient Assisted Living* we have take in consideration his project on an apartment (*"House C"*) disposed on two floors and with only one resident. Its map and its sensors disposition is shown in Figure 4.3, while its supported activities are in the Table 4.2.

Unknown	Leave House	Eating
Brush teeth	Use Toilet Upstairs	Take Bath
Get Dressed	Take Medication	Prepare Breakfast
Get Snack	Get Drink	Put Item in Dishwasher
Put Clothes in Washingmachine	Unload Washingmachine	Receive Guest
Realx	Take Shower	Go to Bed
Store Groceries	Read Paper	Use Toilet Downstairs
Shave	Prepare Lunch	Unload Dishwasher
Watch TV	Prepare Dinner	-

Table 4.2: Kasteren Activity Supported

Figure 4.3: Kasteren House C apartment. On the left there is the first floor, while on the right the second one.

4.1.3 General Interface and Activity Segmentation

After the identification and the analysis of the dataset of interest, the Dataset Collector takes as input the stream of data and parse them in a unique interface accessible by the model and save it in a CSV file. Here, each line corresponds to the corresponding timestamp of the day: for example, the first line denotes midnight and one second (00::00:01),the second line denotes midnight and two seconds (00:00:02) and so on. Instead, the first twenty columns represent the binary status of each sensor, 1 if the sensor is active or 0 if it is off, while the last two columns identify the ID of the activity carried out respectively from the two residents. Moreover, for each timestamp, the entire raw representing the status of the sensors in the house in that moment is called and saved as the *sensor-set* for that precise *timestamp*. Figure 4.4 resume in a better way this scenario.

In addition to the parsing process, this layer manages also the segmentation of data providing the activities of a given day already grouped and ordered by timestamps

Figure 4.4: General parsing interface. In green is highlighted the timestamp column; the red square is the sensor-set while with light-blue and blue identify the activities carried out from the two residents. In the example, at timestamp 216(00:03:36) were active the sensors 4,6 and 19; the resident 1 was doing the activity 12 while the second one the activity 17.

and sensor-sets. In this way the final result is the one shown in the Table 4.3: in the given day of data acquisition it is known that an activity starts exactly at one precise timestamp and it finishes in another; moreover, each sensor-set of this duration (timestemp end-timestamp start) is saved in association with its own timestamp. Resuming, we know when an activity starts, when it finishes and which sensors have been active during this period.

Resident	Activity	Starts	Ends	Sensor-sets
1	Watching TV	0	544	$[ss_1,ss_2,\ldots,ss_{544}]$
1	Talking on the Phone	544	845	$[ss_{544},ss_{545},\ldots,ss_{845}]$
2	Using Internet	0	2623	$[ss_1,ss_2,\ldots,ss_{2623}]$
...	...	...	...	...

Table 4.3: Activity Segmentation of a given day. Each raw in the table identifies which resident (by ID) carries out the given activity, at which timestamp the activity starts, at which timestamp the activity ends and what is the status of each sensor in the house in the corresponding timestamp: for example, ss_{544} is the boolean list of the status of the sensors at 544 seconds (00:09:04) .

4.2 Semantic Layer

Between the data acquisition phase and the one in which the model determines which are the most probable activities carried out by the residents, the core step of the project has been implemented: the **Semantic** Layer. Here, following the guide introduced in Chapter 3, a personal **ontology** on the Activity Recognition field has been designed and developed. At the beginning, it has been really important to analyze the problem from the final objective point of view: it is to delineate the *Behavioral Drift* of a person in a *Smart House*. To do this, it is not so useful to identify the single secondary activity like *Brushing Teeth*, but it is important to note that the person is located in *Bathroom* and that he/she is fulfilling the **Macro** activity *Personal Hygiene*. This is a crucial step because it allows to reduce the wide Activity Recognition domain and it sets the limits to the ontology to avoid exceeding from the real tasks of interest.

4.2.1 Class Hierarchy

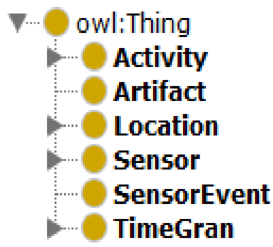


Figure 4.5: Ontology classes

Once the dimension of the problem has been studied and defined, the second step for building an ontology is to identify main classes and their subclasses. Which are the elements always present in the Activity Recognition domain, regardless of the dataset used as input? To answer this question the six entities in Figure 4.5 have been chosen and Figure 4.6 shows all their subclasses:

- *Activity* is the core class of the ontology; it groups up all the activities of interest that can be recognized; notice that each subclass is intended as macro activity, e.g., I do not care if *Breakfast* is *eating* or *preparing*, the important is that the resident is *having breakfast*;

- *Artifact* highlights the objects involved during the fulfilling of a particular activity;

- *Location* identifies an area of the Smart House; for this project only four main rooms of a standard apartment have been identified, but it is possible to add all the needed rooms;

- *Sensor* represents the typology of used sensors; due to the great variance of these types, ARAS legend has been taken as reference as starting point;
- *Sensor Event* is the class that groups up all the events generated by the activation of a single sensor;
- *Time Granularity* divides the day in different phases depending on the time; here, again, only four time granularity phases have been identified but it is possible to add more specific ones like noon or midnight.

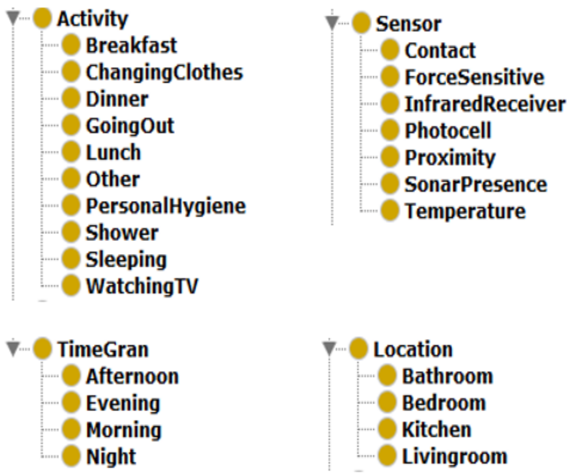


Figure 4.6: Ontology subclasses

4.2.2 Properties Definition

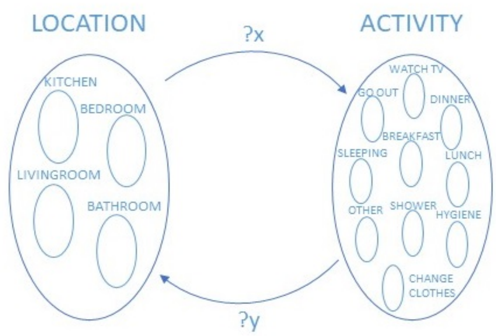


Figure 4.7: Unknown relationships sketch

The class hierarchy design gives us a first sketch on how the ontology is taking form. Once the main elements that compose the domain are defined, the relationships that hold belong them must be generated. At the moment, we are in the situation shown in Figure 4.7: which are the possible and useful relationships that exists between *Location* class and *Activity* class? And between *SensorEvent* class and *Sensor* class? All the possible combinations must be taken into consideration. Moreover, in Chapter 3 we explain that each property has its own **Domain** and **Range**: the first one is the starting class of the property, while the second one is where it ends. Here, for example, *Location* class is the domain for property ?x and *Activity* class is its range; viceversa, *Activity* class is the domain for ?y relationship, while *Location* class is its range. Moreover, if the relationship ?y has exactly the opposite semantic meaning of relationship ?x, ?y could be defined as the **Inverse** property of ?x. This aspect is particularly useful because the Reasoner automatically inferences all the consequent aspects of this semantic relationship: if Shower must occur in Bathroom, the Reasoner already knows that it cannot occur in Kitchen or Livingroom or Bedroom. Table 4.4 sums up all the **object properties** identified for the ontology, specifying which are their domain and range and if they have any inverse; here, a description of each property meaning:

- *canOccurIn*: an activity has its particular locations in which it can be carried out; one person usually have breakfast in the kitchen or in livingroom;
- *hostAct*: a location can host different activities; For example, bathroom can host personal hygiene and having shower;
- *mustOccurIn*: it is similar to canOccurIn, but more binding. For example, personal hygiene must occur in bathroom, it is not possible to have it in livingroom;
- *hasTimeGran*: during which phase of the day can an activity occur? Breakfast usually happens in the morning, while watching TV could occur in each phase of the day (morning, afternoon, evening, night);
- *producesEvent*: one sensor produces its own sensor event;
- *isGeneratedBy*: each sensor event is generated by its respective sensor;
- *isLocated*: one sensor is mounted in a particular room of the house;

- *sensesUsageOf*: an activity, to be carried out, needs some artifacts. For example, sleeping needs a couch or a bed;

- *predictiveSensorFor*: a sensor event could affect the realization of an activity; the sensor event “chair busy” could be predictive for the realization of eating or watching TV;

- *couldBeGeneratedBy*: an activity could be carried out by the acquisition of a particular sensor event; eating could be generated by the sensor events “chair busy” or “kitchen full” or “hall full”;

- *necessarySensorFor*: a sensor event is necessary for the realization of a particular activity; if the TV is turned off it is not possible to watch the TV;

- *mustBegeneratedBy*: to carry out particular activities their necessary sensors must be active during the data acquisition; having shower requires that the door of the shower is closed.

Domain	**PropertyName**	**Range**	**InverseOf**
Activity	canOccurIn	Location	hostAct
Location	hostAct	Activity	canOccurIn
Activity	mustOccurIn	Location	-
Activity	hasTimeGran	TimeGran	-
Sensor	producesEvent	SensorEvent	isGeneratedBy
SensorEvent	isGeneratedBy	Sensor	producesEvent
Sensor	isLocated	Location	-
Activity	sensesUsageOf	Artifact	-
SensorEvent	predictiveSensorFor	Activity	couldBeGeneratedBy
Activity	couldBeGeneratedBy	SensorEvent	predictiveSensorFor
SensorEvent	necessarySensorFor	Activity	mustBegeneratedBy
Activity	mustBegeneratedBy	SensorEvent	necessarySensorFor

Table 4.4: Ontology object properties. Each PropertyName has its class of Domain and Range and in the last column is highlighted if the property is the inverse of another one.

As far as **data properties** are concerned, they are not useful for the purpose of the work because we do not meet particular constraints about data type, but, for future extensions, the property *hasValue* has been identified as an attribute of *SensorEvent* class. Indeed, each sensor-event could assume an *Integer* value of 1 or 0 depending on the status of its sensor.

4.2.3 Individuals

The final step for the realization of the ontology is to *populate* it with instances, known also as **individuals**, of the classes defined in the hierarchy phase. After the creation of an individual of a single class, it will be connected through object properties to other individuals of other classes, defining finally *semantic relations*. For example, in the previous section we have connected *Activity* class with *Location* and *TimeGran* classes thanks respectively to *canOccurIn* and *hasTimeGran* properties. Now, the individual *having dinner* of the subclass *Dinner* has been created and its object properties define that it can occur in *kitchen* or in *livingroom* and that it happens in the *evening*. Figure 4.8 shows this scenario in Protégé: on the left, highlighted in yellow, there is the complete list of the class hierarchy and the *Dinner* class is selected; on the lower left we find in purple the individual *having dinner* created for this class, while on the lower right we find in blue its properties with their respective individuals (*canOccurIn livingroom, canOccurIn kitchen, hasTimeGran evening*). Table 4.5 shows all the individuals created for the Activity classes, to which properties they have been assigned and which are the individuals of the other classes that satisfy these properties. For example, the individual *having breakfast* has been created for the class *Breakfast* and it has three object properties: canOccurIn, canOccurIn and hasTimeGran; the individual of another class that satisfy the semantic relation has been assigned to the respective property: having breakfast canOccurIn in kitchen, canOccurIn in livingroom and hasTimeGran in the morning. The same thing has been done for the Location classes and for the SensorEvent class: Table 4.6 shows the population for Location classes, while Table 4.7 shows the population for SensorEvent class.

4.2.4 Query the Ontology

The ontology development ends with its population and the final graphs grouping up all the entities declared till now is the one of Figure 4.9. Therefore, in the current state, the ontology is ready to be queried and to provide us all useful information that are needed about Activity Recognition domain. Several methods, languages

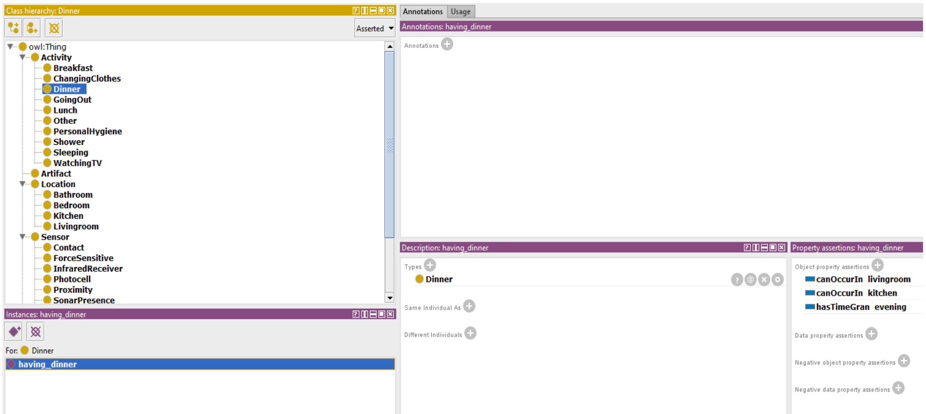


Figure 4.8: Individuals by class and their relationships in Protégé tool. On the left the class Dinner is in yellow and its individual having dinner in purple; its object properties are on the right highlighted by blue.

Class	**Individual**	**Properties**(multi occurrence)	**Target Individuals**
Breakfast	having breakfast	canOccurIn(x2), hasTimeGran	livingroom, kitchen, morning
ChangingClothes	changingClothes	mustBeGeneratedBy, hasTimeGran(x4)	close wardrobe, night, afternoon, evening, morning
Dinner	having dinner	canOccurIn(x2), hasTimeGran	livingroom, kitchen, evening
GoingOut	going out	canOccurIn, hasTimeGran(x4)	livingroom, night, afternoon, evening, morning
Lunch	having lunch	canOccurIn(x2), hasTimeGran	livingroom, kitchen, afternoon
PersonalHygiene	hygiene	mustOccurIn, hasTimeGran(x4)	bathroom, night, afternoon, evening, morning
Shower	having shower	mustOccurIn, hasTimeGran(x4)	bathroom, night, afternoon, evening, morning
Sleeping	sleeping	canOccurIn(x2), hasTimeGran(x4)	bedroom, livingroom, night, afternoon, evening, morning
WatchingTV	watchingTV	mustOccurIn, hasTimeGran(x4)	livingroom, night, afternoon, evening, morning

Table 4.5: Activities population. The first two columns show for each Activity class its individual; indeed, the last two columns show which object properties the individual have (with its occurrences) and which are the individuals of other classes that satisfy the semantic relations. The first row, for example, means that the individual having breakfast of the class Breakfast conOccurIn both in the kitchen and in the livingroom and it hasTimeGran in the morning.

Class	**Individual**	**Properties**(multi occurrence)	**Target Individuals**
Bathroom	bathroom	hostAct(x2)	hygiene, having shower
Bedroom	bedroom	hostAct(x2)	sleeping, changinClothes
Kitchen	kitchen	hostAct(x3)	having dinner, having lunch, having breakfast
Livingroom	livingroom	hostAct(x6)	watchingTV,sleeping,having lunch,having breakfast,having dinner,going out

Table 4.6: Locations population. The first two columns show for each Location class its individual; indeed, the last two columns show which object properties the individual have (with its occurrences) and which are the individuals of other classes that satisfy the semantic relations. The first row, for example, means that the individual bathroom of the class Bathroom hostAct two activities: hygiene and having shower.

and tools exist to extract information from an ontology but for the scope of this project a basic description logic (DL) approach is satisfactory. This is an evolution of first-order-logic that supports all common commands like *and, or, not, exists and so*

Individual	**Properties**(multi occurrence)	**Target Individuals**
bathroom door close	sensesUsageOf, predictiveSensorFor(x2)	door,hygiene,having shower
bed busy	sensesUsageOf, predictiveSensorFor	bed,sleeping
bedroom door close	sensesUsageOf, predictiveSensorFor(x2)	bed door,sleeping,changingClothes
chair busy	sensesUsageOf, predictiveSensorFor(x4)	chair,having lunch,having breakfast,having dinner,watchingTV
close bathroom cabinet	sensesUsageOf, predictiveSensorFor	bath cabinet,hygiene
close kitchen drawer	sensesUsageOf, predictiveSensorFor(x3)	kit drawer,having breakfast,having lunch,having dinner
close tap	sensesUsageOf, predictiveSensorFor	tap, hygiene
close wardrobe	sensesUsageOf, predictiveSensorFor, necessarySensorFor	wardrobe,going out, changinClothes
couch busy	sensesUsageOf, predictiveSensorFor(x2)	couch,sleeping,watchingTV
hall full	predictiveSensorFor(x6)	having breakfast,having lunch,having dinner dinner,watchingTV,sleeping,going out
house door open	sensesUsageOf, predictiveSensorFor	door,going out
kitchen full	predictiveSensorFor(x3)	having breakfast,having lunch,having dinner
open fridge	sensesUsageOf, predictiveSensorFor(x3)	fridge,having breakfast,having lunch,having dinner
oven on	sensesUsageOf, predictiveSensorFor(x3)	oven,having breakfast,having lunch,having dinner
shower close	sensesUsageOf, necessarySensorFor	shower,having shower
tv on	sensesUsageOf, predictiveSensorFor(x4), necessarySensorFor	tv,having breakfast,having lunch,having dinner,watchingTV,sleeping
water closet busy	sensesUsageOf, predictiveSensorFor	water closet, hygiene

Table 4.7: SensorEvent population. The first column shows the individuals name for the SensorEvent; indeed, the last two columns show which object properties the individual have (with its occurrences) and which are the individuals of other classes that satisfy the semantic relations. The first row, for example, means that the individual bathroom door close sensesUsageOf door and it is predicitveSensorFor both hygiene and having shower.

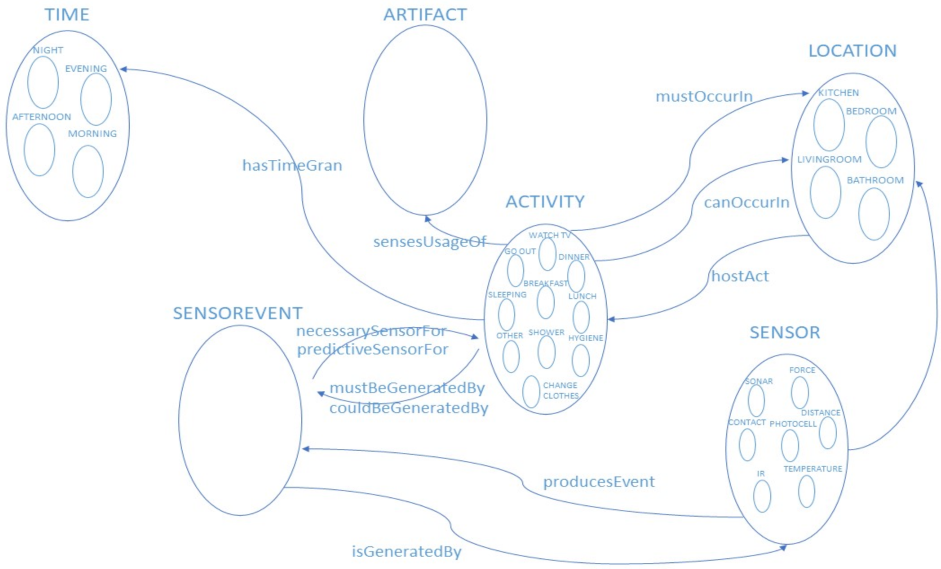


Figure 4.9: Final ontology graph.

on. Protégé already includes an extension that allows to write a DL query and obtain directly the results inside the program. Figure 4.10 shows an example of how it works. To integrate the semantic layer in our model we used OWL api [41], a Java library that manages all the basic operation for the utilization of the ontology like parsing, loading, saving and modifying. Indeed, to query it, a *DLQueryEngine* has been implemented making possible to write a DL query directly by Java console or to define it in the code so that the model can automatically use it. In the example of figure 4.10 we query the ontology with the DL query "((couldBeGeneratedBy value hall full) or (mustBeGenerated value hall full)) and (hasTimeGran value afternoon) and (hasTimeGran value afternoon)"; this query wants as results all that activities that start in the afternoon, end in the afternoon and can be generated by the activation of the sensor event hall full.

4.3 Decision Maker

Taking advantages of Segmentation and Semantic layers it is possible to make the first tests on real datasets. The first step is to load the dataset of interest from the Dataset Collector DB: the first configuration and test happened on ARAS dataset. In this way each day of the acquisition period segmented and formatted in the universal interface explained in the previous sections is available. It is known exactly when an activity of a

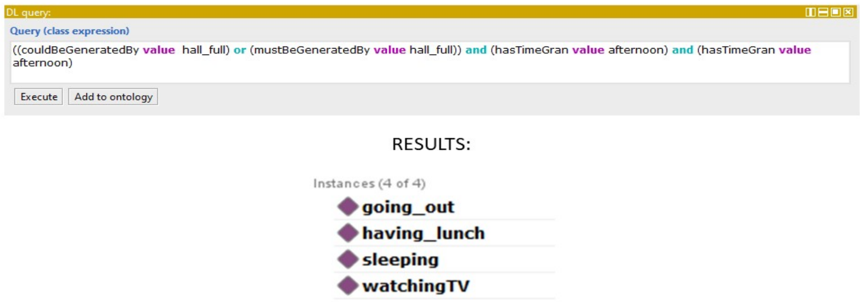


Figure 4.10: DL query example. The DL query "((couldBeGeneratedBy value hall full) or (mustBeGenerated value hall full)) and (hasTimeGran value afternoon) and (hasTimeGran value afternoon)" means all that activities that start in the afternoon, end in the afternoon and can be generated by the activation of the sensor event hall full.

particular day starts, when it ends, where it is located and, the most important aspect, which sensors have been activated during its duration; these sensors are saved in a list called *SensorForActiv*. Before querying the ontology with this useful information it is important to explain the difference between two categories of activities: the ones that the ontology is able to recognize ("activities in the ontology") and the ones that the ontology is not able to recognize ("activities out of ontology"). Why does this distinction exists? The real problem with the "out of ontology" activities is that the dataset is composed by a relative small set of sensors that do not allow us to build a semantic relation strong enough to recognize them; an example could be *Talking on the Phone*: it is impossible to recognize it through the ontology if there is not a sensor mounted on the phone itself. For the moment, these activities are saved in a list *otherAct* and when the model meets one of these activities it ignores them and it sets the Activity Recognition status as *"Recognized"*. In the next section we will explain how the problem of "out of ontology" activities has been managed. Table 4.8 shows this aspect in the case of ARAS dataset: in the first column we find all that activities that our ontology is able to identify, while in the second column there are all that activities tagged as "out of ontology".

Activities in the ontology	**Activities out of ontology**
Having Shower	Washing Dishes
Going Out	Having Snack
Toileting	Studying
Having Breakfast	Napping
Having Lunch	Using Internet
Having Dinner	Reading Book
Sleeping	Laundry
WatchingTV	Shaving
Changing Clothes	Brushing Teeth
-	Talking on the Phone
-	Listening to Music
-	Cleaning
-	Having Conversation
-	Having Guest

Table 4.8: Distinction among ARAS activities. In the first column we find all that activities that our ontology is able to identify, while in the second column there are all that activities tagged as "out of ontology".

At the moment we have all the useful data to query the ontology. The DL query implemented is the same shown in the example of Figure 4.9; for each sensor in the *SensorForActiv* list, the ontology gives back another list of possible activities (*possibleAct*) that comply with the constraints imposed in the query: is the sensor *necessary* for some activities? which activities *could be generated* by the activation of this sensor? which are the activities that could happen during this *time granularity*? Figure 4.11 shows the scenario obtained as result: for each sensor in *SensorForAct*, its *possibleAct* list is obtained by querying the ontology; the Decision Maker entity counts how many times a particular activity appears between the possibleAct lists and it creates a list of identified activities (*identAct*) ordered by their probability occurrence. The starting unknown activity is labeled with the name of the most probable activity of this list.

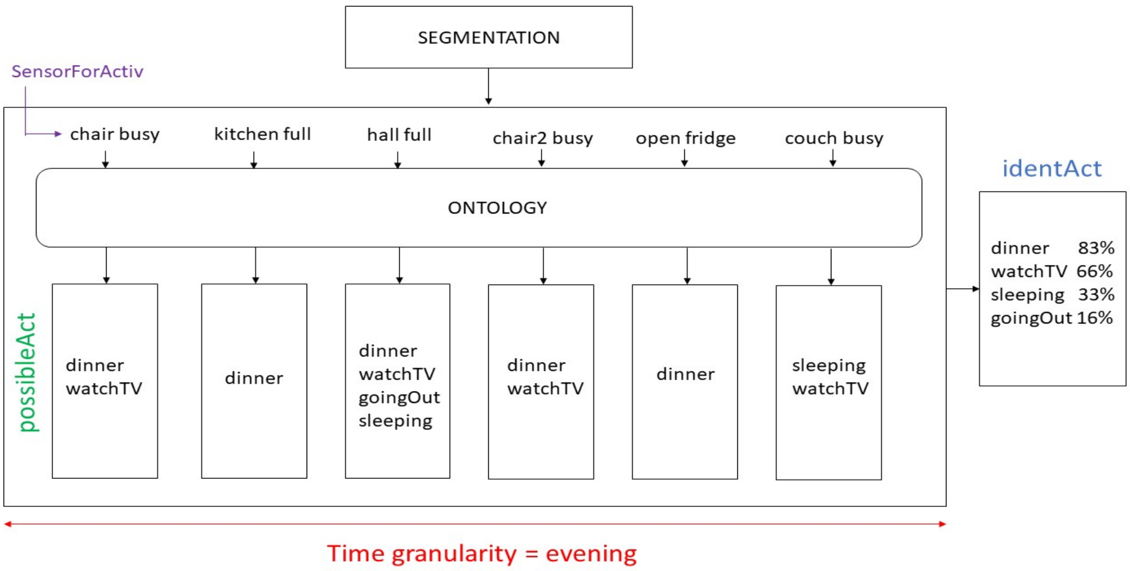


Figure 4.11: Decision Maker first version. Thanks to Segmentation entity we have as input all that sensors (and their relative sensor events) that have been active during the identification of an activity. In this example, SensorForActiv list contains *chair busy, kitchen full, hall full, chair2 busy, open fridge, couch busy*. For each element in the list, the ontology is queried by the DL query shown in Figure 4.9: a new list possibleAct is populated by the activities obtained as result. The activities present in the possibleAct list are the ones that satisfy the constraints of the DL query and they are candidate to be the real activity carried out by the person. Finally, the Decision Maker entity counts how many times a particular activity appears between the possibleAct lists and it creates a list of identified activities (identAct) ordered by their probability occurrence. The starting unknown activity is labeled with the name of the most probable activity of this list (in the example the activity recognized is dinner, with 83% probability).

4.3.1 Filtering and Results on ARAS

Before starting with the first tests on the real ARAS dataset, by analyzing its corresponding CSV files a common problem was noticeable: the noise generated during the data acquisition phase; sometimes a sensor resulted on for one second also if it was not involved in the fulfilment of the activity. This noise made the data really dirty and the corresponding *possibleAct* lists generated by the model resulted too much various and unreal. For this reason **Filter1** has been implemented: a sensor to be saved in the *SensorForAct* list must be activated at least for two seconds during the activity duration. After the application of this filter, a first test on four days of ARAS dataset has been run and the results are shown in Table 4.9.

DAY ID	RECOGNIZED	PERCENTAGE	RECOGNIZED WITH “OTHER”	PERCENTAGE WITH “OTHER”
DAY 1	18/60	30%	43/60	71,66%
DAY 2	18/42	42,86%	34/42	80,95%
DAY 3	26/66	39,4%	54/66	81,81%
DAY 4	24/67	35,82%	58/67	86,56%

Table 4.9: ARAS first results.

Unfortunately these results are too low respect the expectation. Re-analyzing the problem, sometimes the model recognized an activity also if its corresponding necessary sensor was not activated. **Filter2** has been implemented to remove from the *identAct* list these activities: Figure 4.12 shows exactly the same schema of Figure 4.11, but this time the model deletes from identAct list the activity watchTV since its necessary sensor *tv on* was not present among the SensorForActiv sensors.

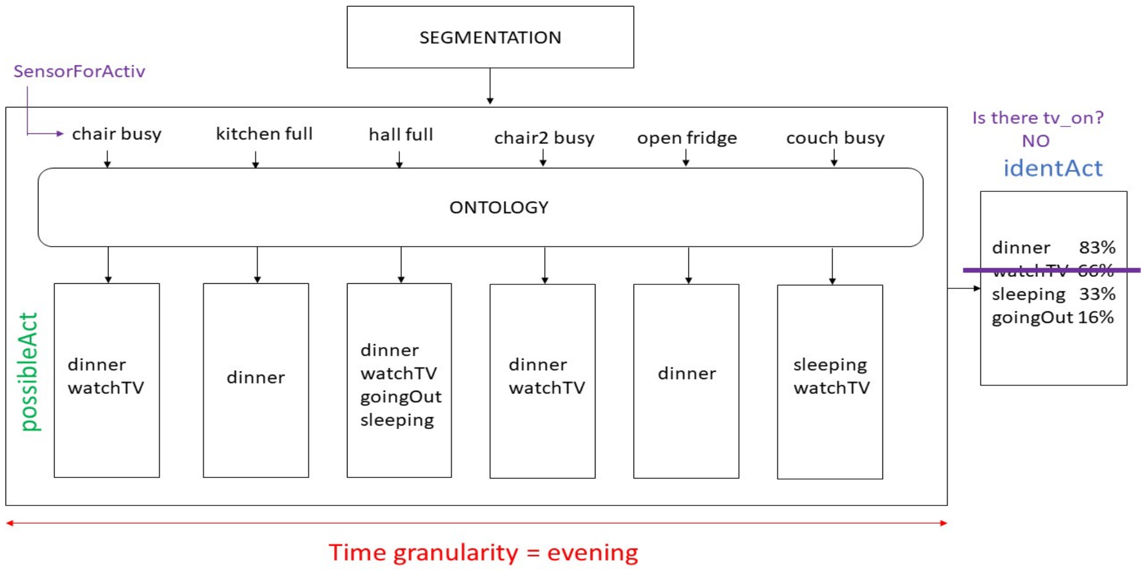


Figure 4.12: Decision Maker second version. The example is the same shown in Figure 4.11, but this time Filter2 delete the candidate activity watchTV by the list identAct because its necessary sensor *tv on* was not present among SensorForActiv sensors.

Moreover, the second resident in the apartment generates another quantity of noise that distorts the final Activity Recognition result. **Filter3**, through a second DL query, saves in SensorForActiv only that sensors that were active in the **Location** where the activity was carried out. In this way, if the second resident activates all the bathroom sensors while the first resident was watching the tv, the model does not risk to recognize *hygiene* instead of *watchTV*. Applying these two other filters, the result obtained are the ones in Table 4.10.

DAY ID	RECOGNIZED	PERCENTAGE	RECOGNIZED WITH "OTHER"	PERCENTAGE WITH "OTHER"
DAY 1	24/60	40%	58/60	96,66%
DAY 2	17/42	40,47%	40/42	95,24%
DAY 3	26/66	39,4%	64/66	96,96%
DAY 4	24/67	35,82%	63/67	94,03%

Table 4.10: ARAS second results.

Activities in the ontology	**Activities out of ontology**
Having Shower	Washing Dishes
Going Out	Having Snack
PersonalHygiene	Studying
Having Breakfast	Napping
Having Lunch	Using Internet
Having Dinner	Reading Book
Sleeping	Laundry
WatchingTV	Talking on the Phone
Changing Clothes	Listening to Music
-	Cleaning
-	Having Conversation
-	Having Guest

Table 4.11: Distinction among ARAS activities after the changes. The table is similar to Table 4.8 with PersonalHygiene instead of Toileting in the first column; this new Macro Activity includes also Shaving and Brushing Teeth that have been deleted from the second column.

The last phase in these filtering process has been the optimization of our ontology with respect to ARAS supported activities. The strength of using an ontology is precisely the possibility of being modified or integrated if needed. It is important to remember that the final scope is to identify the **Macro Activities** carried out by a person inside a smart environment to study its Behavioral Drift. For this reason, it is needless having such activities like *Toileting, Shaving or Brushing Teeth* separated when they could be collapsed under a unique macro activity *PersonalHygiene*. Indeed, in Table 4.11 we can see that PersonalHygiene, with respect to 4.8, takes the place of Toileting, while Shaving and Brushing Teeth have been deleted from the list of the activities "out of ontology". Applying these changes and running the model we obtained the final results shown in Table 4.12.

DAY ID	RECOGNIZED	PERCENTAGE	RECOGNIZED WITH “OTHER”	PERCENTAGE WITH “OTHER”
DAY 1	32/60	53,33%	58/60	96,66%
DAY 2	21/42	50%	39/42	92,86%
DAY 3	33/66	50%	64/66	96,96%
DAY 4	30/67	44,77%	62/67	92,53%

Table 4.12: ARAS final results.

Activities in the ontology	Activities out of ontology
Having Shower	Get Snack
Going Out	Put Clothes in Washingmachine
PersonalHygiene	Relax
Having Breakfast	Store Groceries
Having Lunch	Take Medication
Having Dinner	Get Drink
Sleeping	Unload Washingmachine
WatchingTV	Read Paper
Changing Clothes	Put Item in Dishwasher
-	Receive Guest
-	Unload Dishwasher

Table 4.13: Distinction among Kasteren activities after the changes. In the first column we find all that activities that our ontology is able to identify, while in the second column there are all that activities tagged as “out of ontology”.

4.3.2 Results on Kasteren

ARAS dataset has been the one on which we have taken inspiration to build the ontology. However, once the model has been tested on it, it is useful to understand if it can work well also on different datasets. For this reason we have utilized Kasteren, a dataset with different sensors, different supported activities and only one resident inside the house. After its study and its analysis, we included all the filters implemented for ARAS also in Kasteren. Why should we have to apply also the filter of the sensors

DAY ID	RECOGNIZED	PERCENTAGE	RECOGNIZED WITH “OTHER”	PERCENTAGE WITH “OTHER”
DAY 1	10/14	71,43%	12/14	85,71%
DAY 2	11/15	73,33%	13/15	86,66%
DAY 3	10/14	71,43%	10/14	71,43%
DAY 4	9/9	100%	11/11	100%

Table 4.14: Kasteren final results.

outside the room of interest, if in the house there are only one resident? It is useful to keep also this filter because the data are acquired in a different way from ARAS: the activities acquired during a day are really fewer in number due to the fact that the duration of a single one is instead longer. For this reason, the activity labeled as *hygiene* may contain some sensors like *open fridge* that make the final *identAct* list full of noise. As for ARAS, the ontology has been adapted to Kasteren: starting from the activities in Table 4.2, *Brush Teeth, Shave, Use Toilet Downstairs and Use Toilet Upstairs* have been collapsed under PersonalHygiene; instead *Get Dressed, Leave House, Prepare Breakfast, Prepare Lunch and Prepare Dinner* have been changed respectively in Changing Clothes, Going Out, Having Breakfast, Having Lunch and Having Dinner. After these changes, Table 4.13 shows the activities that the ontology is able to identify with respect to the ones "out of ontology". Applying these changes and running the model we obtained the final results shown in Table 4.14.

4.4 Manage the Others

Till now, we have managed and worked only on the "in the ontology" activities. But how does the system behaves when it meets an "out of ontology" activity that it is not able to recognize? In this section we will show a simple approach to solve this problem. The general idea is that every time the system is not able to recognize a particular activity it asks the resident, through a terminal mounted inside the home, "*which activity are you doing at the moment?*". In this way the resident manually label his/her activity which is saved into the DB together with the list of sensors that were active during the acquisition and its starting and ending timestamps. Every successive times that the system is not sure of the choice, before querying the resident, it checks inside the DB if it has saved already the same list of active sensors with the same start and end time granularity. For example, if the system recognizes the list of sensors (*chair busy, hall full, kitchen full, bed busy*) as an "out of ontology" activity started in the evening and finished in the evening, it verifies if in the DB is already present a labeled activity with the same features (list of sensors and time granularity); if yes, it sets the activity as "recognized", otherwise it asks for help the resident who will manually insert "*Talking on the Phone*". For the realization of these tasks we have created into the DB two new tables organized as follow: Table 4.15 contains the "out of ontology" activity ID "other id", the starting timestamp, the ending timestamp and the activity name (inserted by the resident); instead, Table 4.16 contains the list of active sensors, their personal ID and the "other id" of the relative activity.

OTHER ID	START TIME	END TIME	ACTIVITY NAME
1	night	night	Talking on the Phone
2	night	night	Using internet
3	morning	morning	Studying
4	afternoon	afternoon	Laundry

Table 4.15: First DB table. In the first raw we see that the "out of ontology" activity started in the night and finished in the night has been labeled by the resident as "Talking on the Phone".

SENSOR ID	OTHER ID	ACTIVE SENSOR
1	1	chair busy
2	1	hall full
3	1	kitchen full
4	1	bed busy

Table 4.16: Second DB table. The first four raws of tabel mean that for the activity with "other id" equals to 1 the sensor saved as active were chair busy (ID=1), hall full (ID=2), kitchen full (ID=3), bed busy (ID=4).

Making a JOIN Query on OTHER ID between the two tables we obtain the Table 4.17: for each group of different OTHER ID the system checks if its relative active sensors list is equal to the one under analysis at the moment. The final scope of this approach is to populate the DB enough to reduce as much as possible the interaction between the resident and the terminal.

OTHER ID	ACTIVE SENSOR	SENSOR ID	ACTIVITY NAME	START TIME	END TIME
1	chair busy	1	Talking on the Phone	night	night
1	hall full	2	Talking on the Phone	night	night
1	kitchen full	3	Talking on the Phone	night	night
1	bed busy	4	Talking on the Phone	night	night
2	tv on	5	Using Internet	night	night
2	couch busy	6	Using Internet	night	night
2	chair busy	7	Using Internet	night	night
2	hall full	8	Using Internet	night	night
2	kitchen full	9	Using Internet	night	night
2	bed busy	10	Using Internet	night	night
3	...	...	...	...	...

Table 4.17: Final DB table. For each OTHER ID group, the system checks if it active sensor list is the same as one already present in the DB.

Chapter 5

Conclusions

This work shows how it is possible to apply and to take advantage of the *Semantic Web*, and more generally of the *Semantics*, for giving support to an unsupervised model able to recognize *Activities of Daily Living*. The first step to reach this goal has been a substantial analysis and research on how the Semantic world could be useful in the Activity Recognition field and which are its potentialities. The crucial point of the entire project has been the design and development of an *Ontology*, a Semantic vocabulary holding all the useful information of a domain of interest. Taking inspiration from the ARAS dataset model, we define our personal *Class Hierarchy* and the Semantic (*Object*) *Relations* that exist among their *Individuals*. The main features identified for doing Activity Recognition have been the *Location* in which the activity happens, in which phase of the day (*Time Granularity*) it is usually carried out and which is the *list of sensors* that were active during its recognition process. The combination of these information with a segmentation phase that provides the list of active sensors related to a single activity allows us to query the ontology through Description Logic (DL) queries, obtaining as a result the most probable activity carried out by the resident inside the Smart Home. Finally, we have tested our model on two different datasets, ARAS and Kasteren, during four days of observations: for ARAS the model is able to recognize correctly the 94,75% of the activities carried out by the resident, while for Kasteren the 85,95%. These results, although satisfactory, are not real. Indeed, in this first version, the system sets as recognized also that activities that the ontology is not able to recognize: for example, without a sensor on the telephone, it can not know that the resident is talking on the phone. To solve this problem, we extend our model with an approach that involves directly the resident: whenever the system meets an activity that it is not able to recognize, the resident is asked to manually insert, through a terminal mounted inside the home, which activity he/she is currently doing. His/her answer will be inserted into the DB together with the list of sensors that were active during the acquisition and its starting and ending timestamps. Now the system, every time that meets an "out of ontology" activity, checks if into the DB is already present a labeled activity with the same features; if yes, it sets the activity as recognized, otherwise it asks for help the resident. The final scope of this approach is to populate the DB enough to reduce as much as possible the interaction between the resident and the terminal. We tested this approach populating the DB with 10 days of observations in ARAS, obtaining as a result for the 11th the 91,43% of recognized activities, instead of 41,42%.

5.1 Future works

The focus of this thesis has been principally on the Semantic layer of the model: its adaptability feature allows it to be applied to many different methods regarding the Activity Recognition field. Indeed the ontology, depending on the need, could be manually extended or modified. The development of a solid probabilistic algorithm to be joined to our ontology could be certainly an important improvement for the system: actually, an activity is recognized only thanks to its occurrences. Moreover, the main limit of this project regards the step of recognizing the "out of ontology" activities: novel approaches like *Active Learning* or *Deep Learning* could be useful for this application.

Bibliography

[1] K Mack and L Thompon. Data profiles, family caregivers of older persons: Adult children.(2001). *Georgetown University, the Center on an Aging Society.*

[2] Marta Inés Berrío Valencia. Aging population: a challenge for public health. *Revista Colombiana de Anestesiología*, 40(3):192–194, 2012.

[3] Fabio Veronese, Hassan Saidinejad, Sara Comai, and Fabio Salice. Elderly monitoring and aal for independent living at home: human needs, technological issues, and dependability. In *Optimizing Assistive Technologies for Aging Populations*, pages 154–181. IGI Global, 2016.

[4] Simone Mangano, Hassan Saidinejad, Fabio Veronese, Sara Comai, Matteo Matteucci, and Fabio Salice. Bridge: Mutual reassurance for autonomous and independent living. *IEEE Intelligent Systems*, 30(4):31–38, 2015.

[5] Assistive technology group (atg) of politecnico di milano. http://www.atg.deib.polimi.it, 2015.

[6] Unsupervised methods for activities of daily living drift modeling and recognition. http://hdl.handle.net/10589/116330, 2015.

[7] Fabio Veronese, Daniele Proserpio, Sara Comai, Matteo Matteucci, and Fabio Salice. Sharon: a simulator of human activities, routines and needs. *Stud. Health Technol. Inform*, 217:560–566, 2014.

[8] Richard Harper. *Inside the smart home.* Springer Science & Business Media, 2006.

[9] Diane J Cook. How smart is your home? *Science*, 335(6076):1579–1581, 2012.

[10] Martin Becker, Ewoud Werkman, Michalis Anastasopoulos, and Thomas Kleinberger. Approaching ambient intelligent home care systems. In *Pervasive Health Conference and Workshops, 2006*, pages 1–10. IEEE, 2006.

[11] Carles Gomez and Josep Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6), 2010.

[12] Yang Hao and Robert Foster. Wireless body sensor networks for health-monitoring applications. *Physiological measurement*, 29(11):R27, 2008.

[13] Diane J Cook, Narayanan C Krishnan, and Parisa Rashidi. Activity discovery and activity recognition: A new partnership. *IEEE transactions on cybernetics*, 43(3):820–828, 2013.

[14] Jessamyn Dahmen, Brian L Thomas, Diane J Cook, and Xiaobo Wang. Activity learning as a foundation for security monitoring in smart homes. *Sensors*, 17(4):737, 2017.

[15] Narayanan C Krishnan and Diane J Cook. Activity recognition on streaming sensor data. *Pervasive and mobile computing*, 10:138–154, 2014.

[16] Diane J Cook. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 27(1):32–38, 2012.

[17] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2013.

[18] Daniele Riboni and Marta Murtas. Web mining & computer vision: New partners for object-based activity recognition. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2017 IEEE 26th International Conference on*, pages 158–163. IEEE, 2017.

[19] Yonggang Lu, Ye Wei, Li Liu, Jun Zhong, Letian Sun, and Ye Liu. Towards unsupervised physical activity recognition using smartphone accelerometers. *Multimedia Tools and Applications*, 76(8):10701–10719, 2017.

[20] David Bannach, Martin Jänicke, Vitor F Rey, Sven Tomforde, Bernhard Sick, and Paul Lukowicz. Self-adaptation of activity recognition systems to new sensors. *arXiv preprint arXiv:1701.08528*, 2017.

[21] KS Gayathri, KS Easwarakumar, and Susan Elias. Contextual pattern clustering for ontology based activity recognition in smart home. In *International Conference on Intelligent Information Technologies*, pages 209–223. Springer, 2017.

[22] Juan Ye, Graeme Stevenson, and Simon Dobson. Usmart: An unsupervised semantic mining activity recognition technique. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(4):16, 2015.

[23] KS Gayathri, KS Easwarakumar, and Susan Elias. Probabilistic ontology based activity recognition in smart homes using markov logic network. *Knowledge-Based Systems*, 121:173–184, 2017.

[24] Daniele Riboni, Timo Sztyler, Gabriele Civitarese, and Heiner Stuckenschmidt. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1–12. ACM, 2016.

[25] Gabriele Civitarese, Claudio Bettini, Timo Sztyler, Daniele Riboni, and Heiner Stuckenschmidt. Nectar: Knowledge-based collaborative active learning for activity recognition. 2018.

[26] Casas website. http://casas.wsu.edu/.

[27] Mathieu Gallissot, Jean Caelen, Nicolas Bonnefond, Brigitte Meillon, and Sylvie Pons. *Using the multicom domus dataset*. PhD thesis, LIG, 2011.

[28] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International conference on pervasive computing*, pages 158–175. Springer, 2004.

[29] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. 2001.

[30] The world wide web consortium (w3c). https://www.w3.org/Consortium/.

[31] Making the web accessible. https://www.w3.org/WAI/.

[32] Web of devices. https://www.w3.org/standards/webofdevices/.

[33] Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of xml and rdf. *IEEE Internet computing*, 4(5):63–73, 2000.

[34] Resource description framework. https://www.w3.org/RDF/.

[35] Vocabularies. https://www.w3.org/standards/semanticweb/ontology.

[36] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[37] Joanne S Luciano, Bosse Andersson, Colin Batchelor, Olivier Bodenreider, Tim Clark, Christine K Denney, Christopher Domarew, Thomas Gambet, Lee Harland, Anja Jentzsch, et al. The translational medicine ontology and knowledge base: driving personalized medicine by bridging the gap between bench and bedside. In *Journal of biomedical semantics*, volume 2, page S1. BioMed Central, 2011.

[38] Deborah L McGuinness, Richard Fikes, James Rice, and Steve Wilder. An environment for merging and testing large ontologies. In *KR*, pages 483–493, 2000.

[39] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[40] Protégé tool. https://protege.stanford.edu/.

[41] Owl api. http://owlcs.github.io/owlapi/.

[42] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. Aras human activity datasets in multiple homes with multiple residents. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare*, pages 232–235. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013.

[43] Owl api. https://sites.google.com/site/tim0306/datasets.