Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Informatica, gestionale e dell'automazione

Smart Environment Design and User's Behaviors Mining in AAL Domain

Ph.D. Dissertation of:
Marco Cameranesi

Advisor:
prof. Domenico Potena

Curriculum Supervisor:
prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell’Ingegneria
Curriculum in Ingegneria Informatica, gestionale e dell’automazione

---

Smart Environment Design and User’s Behaviors Mining in AAL Domain

Ph.D. Dissertation of:
Marco Cameranesi

Advisor:
prof. Domenico Potena

Curriculum Supervisor:
prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...'

Isaac Asimov

Acknowledgments

After an intensive but interesting period, today is the day: writing such words of thanks is the final destination of this journey. It has been a period of intense learning for me, not only in the scientific arena, but also on a personal level. I wish to thank my family and friends for their significant and daily support. Thanks to Max, Gauss and Fabio for their sincere friendship and being always present when needed. I would also like to express my gratitude to all the guys of the White Rabbit Hole for their friendship. Last but not least, I'd like to thank my supervisor Domenico, that taught me so much guiding me through this three years research experience.

Ancona, November 2017

Marco Cameranesi

Abstract

In recent years, more and more efforts were put in the design and development of Ambient Assisted Living (AAL) solutions. This research is focused on applications of semantic web and data mining techniques to the field of AAL. In particular the activities concern two major aspects. The first one is related to solutions aimed at supporting the design of AAL systems. Indeed, the introduced methodology that, starting from designer's requirements, provides a set of sensors which allow to achieve the project objectives. In order to fulfill such a task, a goal-driven ontology has been developed and on its top a reasoning framework is introduced. The ontology includes three different modules. The Goal module which describes the decomposition of a goal in sub-goals and tasks. The Measure module which is aimed at describing the measures and their relations with all the objects belonging to the modeled domain. Finally the Sensor module which defines the sensors and their characteristics. The Measure module, which is based on the introduction of a novel ontology, that describes the mathematical relations and aspects, represents a relevant contribution of this work. For example, given a certain goal it can be decomposed in one or more sub-goals and tasks. Starting from such elements, the needed measures to achieve them can be defined. Through mechanisms related to mathematical manipulation formulas, which are enabled by the Measure module, the required measures are extracted and then properly combined together in order to provide the final measures required to achieve the goal. The second aspect which is the objective of this research, is related to Process Mining techniques aimed to study the behavior of a user that lives in an AAL environment. In particular, taking advantage of process discovery algorithms which are able to extract the model of a process, starting from process event logs. Such algorithms can be adopted in contexts in which the process model is unknown or they can be helpful to check the conformance of the actual model to the design-time model. In the AAL field such algorithms have been used on data collected through sensors of a smart environment. After a proper pre-processing stage, data have been exploited to extract two kinds of different models. The daily behavioral model which represents the behavior of a standard day of the user, whereas the macro activity models which are a set of models representing the flow of sensors activations when the user perform a given daily activity (e.g. watching the tv, cooking, reading and so forth). The obtained models have been verified

with interesting results through conformance checking algorithms. In the Literature approaches exist that exploit process mining techniques to define the daily behavioral model, the methodology for the extraction of the macro activity models is something novel and represents the most original contribution of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Smart Technologies

Nowadays the research in the field of information communication technologies (ICTs) is constantly improving. In particular, the diffusion of advanced electronic devices is facilitated by low manufacturing costs. As technology is progressing, computational speed and efficiency are regularly increasing.

In this evolving scenario embedded platforms play a significant role. The development of ever smaller electronic devices and components has brought to embed them in familiar objects. We can just think about smartphones, laptops, home appliances (e.g. refrigerators, washing machines, microwave ovens) and so on. Since such a fact, the vision of the ubiquitous computing paradigm [1] becomes a reality. Ubiquitous computing, also known as pervasive computing indeed focuses on the capabilities of objects to communicate and perform helpful tasks aimed at minimizing the interaction of end users with traditional computers. In the ubiquitous computing paradigm there is an interaction amongst various actors: devices, sensors and the surrounding environment, as depicted in Figure 1.1.

These smart objects can be easily connected to each other through local and global network structures that can be found almost everywhere. In a world where the above mentioned technologies have become real, the applications that take advantage of them, represent the so called Internet of Things (IoT) [2]. Asthon was the first researcher to introduce the term Internet of Things and in the following lines his words are reported to describe such a concept: “we need to empower computers with their own means of gathering information, so they can see, hear and smell the world for themselves, in all its random glory” [3]. The IoT is much more than a simple technology, in fact it expresses an innovative paradigm representing a new phase of the Internet evolution. Such a trend is increasing, the European Commission estimated that in 2020 the devices connected to the Internet will range between 50 and 100 billion [4].

The universAAL specification [5] provides a definition for the smart space concept. It can be considered as “an environment centered on its human users in which a set of embedded networked artifacts, both hardware and software, collectively realize the paradigm of ambient intelligence”. Such a definition

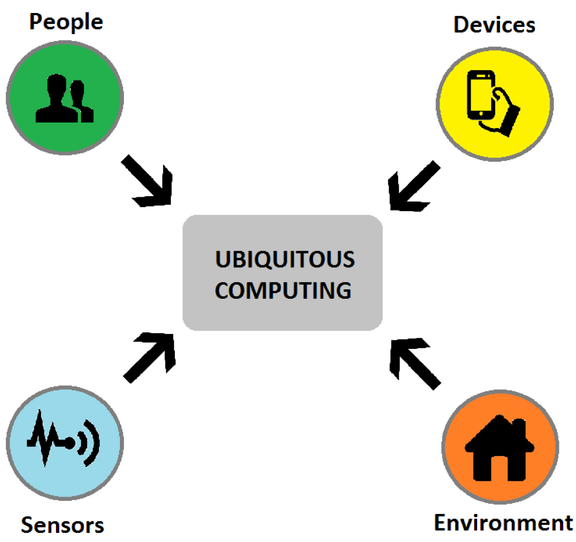

Figure 1.1: Ubiquitous computing interactions.

directly points to the concept of Ambient Intelligence (AmI). According to Aarts and Encarnação in [6], AmI "refers to electronic environments that are sensitive and responsive to the presence of people" and "in an AmI world, devices operate collectively using information and intelligence that is hidden in the network connecting the devices" they "cooperate seamlessly with one another to improve the total user experience through the support of natural and intuitive user interfaces".

As claimed by Cook et al. [7] AmI systems are characterized by five key technology features: sensitivity, responsiveness, adaptivity, transparency and ubiquity. Sensitivity highlights the ability of the system to sense the surrounding environment, and understanding the context. Adaptivity and responsiveness are related to the capacity of reacting to the ambient changes. Ubiquity and transparency are two concepts closely associated to the pervasive computing paradigm envisioned by Mark Weiser [1] who stated that: "The most profound technologies are those that disappear. They wave themselves into the fabric of everyday life until they are indistinguishable from it". Starting from previous definitions, we can introduce the assisted living technologies, which are technological solutions relying on Ambient Intelligence and that are known as Ambient Assisted Living systems. A general definition of the Ambient Assisted Living paradigm can be considered as the use of information and communication technologies (ICT) in a person's daily living and working environment which is aimed at allowing people to keep healthy, live safely and independently. Ambient Assisted Living systems can be exploited for a variety of useful applications that can be classified depending on their main objective as follows:

- *anomaly detection:* it concerns the identification of events, items or patterns which don't match an expected behavior. In the Literature over the years many anomaly detection techniques have been proposed, mainly based on clustering techniques, statistical method or others. In AAL systems, anomaly detection is widely adopted to detect possible anomalies in daily activities, to alert people about dangerous situations or even to check if any external help is necessary [8]. Jakkula et al. in their work [9] propose a methodology based on temporal relation discovery with the aim to detect anomalies within a smart environment. The authors state that anomaly detection is precise if based on frequent and predictable behaviors. Thus they look for temporal interactions only among the most frequent performed activities. Furthermore, this approach let them to reduce computational costs.

- *planning reminders / schedulers*: another relevant aspect of AAL systems is represented by automatic planning platforms. For example, planning permits to schedule plans and create daily reminders. Such solutions are mainly aimed to help people with memory impairment to live safer and more independently [10]. In general one of the major challenges of such systems is when to remind and what to remind. The key aspects on which researchers focus are: to avoid alerting too many times, to avoid alerting at inappropriate times, to manage task priorities in smart ways and to prompting the user at the right time, i.e. not too soon or not too late. In a more advanced approach [11] the authors propose a complex system where the planning can be dynamically learned or adapted based on observations. They show how adaptive options can quickly adapt their behavior depending on the way the user acts.

- *security - indoor location system identity / management:* indoor location systems represent another substantial component in AAL systems. Indoor positioning systems provide a way to locate objects or people within a given environment by the use of magnetic fields, radio waves, acoustic signals or other information gathered by mobile devices [12]. Such systems represent a valid solution in order to improve the security of a smart environment. In [13] Xiong et al. present a system that uses MIMO-based techniques to track wireless clients as they move within a closed space like a building. The localization system has been validated with up to forty clients and is able to detect client positions with a good degree of spatial and temporal spatial accuracy. Another different approach proposed by

Orr and his research team is based on the ability to detect the presence of users by using pressure sensors located on the floors [14]. Further common ways to detect user presence are based on PIR (Passive Infra-Red) sensors which can detect motion when an infrared source (such human beings) passes through the sensor field. An important aspect related to the localization problem, consists in the identification process of a person who is currently walking on a room or taking some medications. To distinguish inhabitants we can refer to two general approaches. The more accurate active identification methods take advantage of technologies like RFID tags to identify people [15], whereas the anonymous approach exploits machine learning techniques in order to build motion models of every person [16], [17].

- *health monitoring:* in Ambient Assisted Living the monitoring process of relevant vital parameters of people is also a significant task. A large number of AAL systems are intended to monitor daily activities with the aim of preserving independent living, automated assistance and remote health monitoring. Some researchers focus their effort on monitoring a single activity or parameter whereas others multiple ones. Specifically in the last years many researchers presented a wide range of systems devoted to remotely check the health of people. The scientific community generally refers to such systems with the term telemedicine [18]. AMON [19] is a telemedical monitor developed at the ETH Zurich University. The presented system is able to gather data in order to check multiple vital signs, it can measure blood pressure, SpO2 (peripheral oxygen saturation level) and it incorporates a lead ECG. This device is also provided with an on-board GSM transceiver which allows data exchange with medical centers.

  Progresses in such field have been done also thanks to the miniaturization processes and the material engineering. In fact wearable and e-textile specific sensors have been developed with the purpose of monitoring important health parameters. Examples related to the above mentioned technologies are reported in [20] and [21].

  Fall detection is another significant topic of health since falls are an important cause of mortality of aged people. Wu et al. [22] propose a Smart Cane which is able to analyze its usage, informing the user of possible falling risks and various kind of statistical parameters. Furthermore some researchers proposed also wearable fall detection systems which are able to detect the user's posture and motion taking advantage of accelerometers and gyroscopes [23], [24]. A complementary approach relies on ambient fall detection systems which are based on PIR or floor pressure sensors [25] and audio analysis [26].

- *power saving / energy management:* a home energy management system (HEMS) is usually part of a smart environment. Such systems are aimed at handling in an intelligent way the power consumption in order to reduce costs of energy monitoring and optimizing the power consumption of devices and domestic appliances.

  Krishna et. al [27] present in their paper a project based on Power Line Communication (PLC) system. Their solution provides a real time access to information on home energy consumption. Moreover the system is also able to control appliances and optimize power consumption. The platform consists of three interoperating modules: an advanced power control planning engine, a device control module and a power resource management server. Other approaches like [28] exploit wireless protocols in order to build a platform aimed at balancing power loads and controlling energy consumption. The adoption of wireless technology leads to relevant advantages in terms of scalability and possibility to easily and quickly connect mobile devices. After reviewing some wireless technologies like WiFi protocols (802.11/a/b/g), Bluethooth, UWB (802.15.3) and ZigBee the researchers decided to focus on ZigBee due to its high level of efficiency. The authors also highlight that such system is appropriate to reduce implementation time and costs. Cottone et al. [29] point out how energy saving in home environments can be achieved taking advantage of activity recognition techniques. Indeed their research is based on the detection of users common activities in order to predict the trend of energy consumption. To this aim the system controls the working times of appliances depending on the user habits which are recognized from raw sensor data exploiting an iterative a priori-based algorithm called Discontinuous Varied-order Sequential Miner (DVSM) [30].

1.1 Sensors and actuators: the physical layer

In this section it is provided a quick overview of a significant aspect of each Ambient Assisted Living systems, that is the physical layer. The electronic devices, during the last few years, have benefitted from quick and relevant advancements of the research in the field of electronics. In relation to the Ambient Assisted Living domain, the electronic devices can be briefly classified as follows.

- *ambient sensors:* context information [31] can be retrieved by analyzing the data gathered from the different sensors within a smart environment.

Smart homes exploit such knowledge for several kind of automation purposes, ranging from providing a high comfort level for the inhabitants to preventing dangerous situations. The most used sensors for this kind of applications can be summarized in the following way:

- *motion sensors:* in general, motion can be considered a position's change of objects or persons with respect to time and their initial reference location. Motion can be characterized in terms of position displacement, speed, acceleration and time. Motion sensors are then aimed to detect and track movements within a monitored environment. There exist many types of motion sensors that can be classified depending on the technology.

  PIR sensors: are able to detect the infrared radiation emitted by the human body. This way a system exploiting such sensors can estimate the position and the velocity of a person. For instance, Popescu et al. [32] proposed a vertical PIR sensor array system aimed at detecting falls. With the aim of differentiating falls to the other human activities (non-fall) the authors have taken advantage of a pattern recognition algorithm based on Hidden Markov Models.

  Ultrasonic sensors: which generate pulses of ultrasonic waves and measure the reflection of a moving object. A possible inconvenience of ultrasonic sensors is that the sensor can sense the motion within undesired areas, for example, because of reflections of sound waves around corners.

  Microwave sensors: which emit frequency in the microwave region so that analyzing the returned and/or reflected microwave frequency the system is able to detect motion.

  Tomographic sensors: such a type of detector works creating a mesh network of radio waves which ignore obstacles like walls. Disturbance in the network allow to detect motion over the area covered by the network.

- *Radio Frequency Identification (RFID) sensors:* they take advantage of electromagnetic fields to automatically identify and track electronic tags which are attached to objects. RFID are commonly used in many fields, for instance, an RFID tag attached to an object during the various production steps can be adopted to check its progress through the assembly line. There are platforms aimed at distinguishing users through badges built on such a technology. For instance we can refer to the Landmarc [33] and SpotOn [34] projects.

  - *Pressure sensors / smart tiles:* can be employed to retrieve information about the pressure which is applied on something. For example smart tiles can detect the footsteps of a user, whereas other pressure sensors can be exploited to check if the user is on the bed or is sitting on a chair.
  - *Cameras:* which are basically aimed at recording video streams. Nowadays video processing algorithms are able to detect changes in the surrounding context helping to identify the user activities. Moreover they can be profitable for safety and security purposes, indeed they can be employed to detect and prevent dangerous situations and anomalies.
  - *Acoutstic sensors:* sound and audio processing techniques can be exploited for the activity recognition purposes. For example, Huang et al. [35] in their work propose a system, based on Hidden Markov Model (HMM), which is able to detect events according to the revealed audio signals coming from microphones. Vuegen et al. [36] used low-power Wireless Acoustic Sensor Network (WASN) to recognize daily living activities in a real case scenario. The system exploited also ultrasound waves in addition to the audio to improve activity recognition.
  - *magnetic sensors:* they can be exploited to detect the status of doors, windows or drawers. For example if installed on the external doors of a house, they allow to know when someone is leaving or coming back home. If this kind of sensor are used in combination with RFID tags, they can give information about the identity of the person that is opening a door or a specific drawer.

- *wearable and mobile sensors:* recent advances in microelectronics and sensor manufacturing have led new relevant scenarios regarding wearable technologies for remote monitoring. Miniaturization processes have improved and several technologies like MEMS (Micro Electro-Mechanical Systems) have been developed in recent years. In order to monitor health of people, today non-intrusive sensors are available in the form of patches, small devices, body-worn devices, and smart habiliment accessories. Wearable sensors allow monitoring and recording real time data about physiological conditions and motion with a very reduced manual activity. Nowadays wearable sensors can be found widespread, for instance the majority of smartphones is equipped with a GPS (Global Positioning System), a gyroscope and an accelerometer. As an example, they can be profitably employed for monitoring the mobility and the activity of a person. In the Literature there is a number of relevant works based on

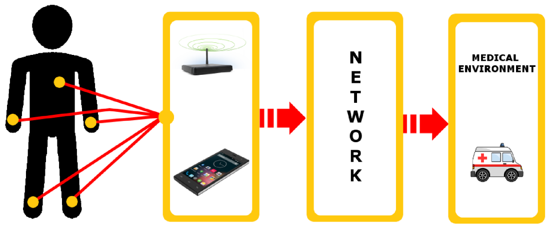

Figure 1.2: Body Area Network architecture example.

this concept. Bayat and his research team in [37] show how it is possible to detect a set of daily activities, in real life conditions, starting from the data gathered by triaxial accelerometers of a smartphone. Their approach offers a solution to the problem, by applying supervised classification algorithms. The training data is retrieved experimentally thanks to real people performing a set of prearranged activities. In their work they focus on the robustness of the proposed system and on the reduction of the number of necessary sensors keeping at the same time a good level of accuracy. Other relevant contributions comes from the field of *Epidermal Electronics* [38] which is nowadays considered as a cutting-edge discipline in which electronic circuits are perfectly integrated to the surface of the human skin in a way that is mechanically transparent to the user. Taking advantage of such an approach, a new type of tattoo-like electronic devices is going to replace conventional wearable pads, wrist-band and smart watches in the near future.

Mobile sensors are usually connected together through a network architecture known as body area network (BAN) [39] which grant wireless communication around the human body, see Figure 1.2.

The measurements retrieved by sensors are then transmitted to a processing station adopting convenient communication protocols. Since often the processing stations are near, it is preferable to use low power communications protocols like ZigBee, ANT, Bluethooth for energy saving purposes [40]. The processing station can be either a PDA (Personal Digital Assistant), a smartphone a computer or another platform where processing information algorithms are executed.

1.2 Smart space applications

During these last years several Universities and Research Institutes all over the world have invested time and resources to build real smart spaces in order to test their technologies. Such environments can be considered a sort of laboratories where users and researchers can extensively experiment and evaluate the newest solutions regarding smart environment technologies. Amongst them it is worth citing the most representative and relevant projects.

The CASAS Project [41] provides a non-intrusive assistive environment for monitoring patients affected by dementia at home. The architecture is composed of a physical layer which include the hardware components like sensors and actuators which communicate through the ZigBee protocol. Moreover there is a software layer that exploiting support vector machine (SVM) algorithms deal with the real-time activity recognition. In order to evaluate the ability of the models they have gathered data from sensor in several different smart homes. Data were manually annotated in order to provide activity labels. The results show that some activities are easier to recognize than others, since some of them overlap each other.

The University of Colorado developed a smart home [42]. The apartment is equipped with more than seventy-five sensors and actuators. They are devoted to monitor several aspects of the environment, such as rooms temperatures, ambient light, sound level, motion, door and window positions. Moreover they monitor even the external part of the house in order to verify weather and insolation conditions. Actuators are responsible to manage air and water heating, lighting, and ventilation.

The Gator Tech Smart Home [43] is a live experimental laboratory for validating technologies and systems developed in the Mobile and Pervasive Computing Laboratory of the University of Florida. Helal and his team focused their efforts in activities aimed to assist aged people and individuals with special needs. The aim was giving to them independence guaranteeing high quality life standards. The Gator Tech Smart Home (GTSH) is a single-family apartment used as a testing platform to implement and validate various smart solutions. The smart home consists of a multi layer architecture. Starting from a low level physical layer represented by the single electronic devices and appliances to the more abstract Knowledge Layer, devoted to represent through the adoption of an ontology all the services offered to the users and all the connected devices.

Yamazaki and his research team developed the Ubiquitous Home [44], which represents a space to experiment advanced context aware services. The project principally aimed at recognizing the situations of the inhabitants and focusing on their contextual information in a real life situation. The home is provided with several cameras and microphones, moreover in each room there are several kind of sensors aimed to monitor every space. The apartment is composed of a living room, a dining-kitchen, a study, a bedroom, a washroom and bathroom. Each room has cameras and microphones in the ceiling to collect video and audio data. There are also floor pressure sensors which are installed with the aim of tracking the residents and detect furniture positions. The apartment is also equipped with infra-red sensors in order to detect the users motion.

The Massachusetts Institute of Technology (MIT) has developed smart home projects within the House n department [45]. The aim of their project is testing new materials, technologies and strategies focused on exploring activities and interactions of people within smart environments. The most relevant initiative comprises The PlaceLab which can be considered a living laboratory, used to study technology and design strategies. Hundred of sensors are installed in each part of the home and are employed to support the design of innovative user interface applications aimed at helping people to control their environment, save resources, keep active and healthy. The sensors are also exploited to check the activity within the environment so that researchers can analyze how people react to new devices, systems, and architectural design strategies in the complex context of the home.

1.3 Our contribution

The main contribution to the field of Ambient Assisted Living can be considered twofold. Several knowledge models have been proposed in the last years in order to describe Abient Assisted Living sensors and actuators. At the moment models and methodologies aimed at providing a complete assistance to designers during the development of an AAL environment are still missing. To this end, in the first part of this work it is introduced an ontology which is devoted to represent in a formal manner, all relevant aspects related to knowledge in the AAL domain, ranging from goals to measures and sensors. A set of logic-based reasoning functions which give advanced support to the design process are also defined.

With the introduced solution, starting from high-level goals AAL designers can easily retrieve which devices are needed to best fit the project requirements. It directly and efficiently lead to a cost-effective reduction of development time.

The second aspect of this work focuses to exploit smart environment sensors data. In particular, smart environments improve the life quality of people, providing inhabitants with advanced functionalities and services that support them during their daily activities. As previously explained, in order to implement such improvements, smart environments are equipped with many sensors that constantly monitor the activities performed by a user. Sensor data are activation sequences and can be considered as the execution of a process representing daily user behaviors and performed activities. In the second part of this dissertation a methodology is introduced, aimed at taking advantage of Process Mining techniques in order to discover models representing and characterizing the user behaviors.

This work is structured as follows: in Chapter 2 it is introduced a sensor network goal-driven ontology, building on its top a reasoning framework. Chapter 3 introduces a methodology capable to exploit sensors' data in order to extract models describing the behavior of a smart environment user. The proposed methodology is also supported and evaluated thanks a real case study. Chapter 4 details about other Ph.D. activities related to Educational Data Mining approaches aimed at discovering models suitable to represent student's careers.

Chapter 2

The GoAAL ontology

2.1 Introduction and motivation

In recent years, a variety of assistive technologies have been proposed to improve people's life quality thanks to environments that are sensitive, adaptive, and responsive to human needs. Relying on this concept, Ambient Assisted Living (AAL) technologies, exploit sensors and actuators for several monitoring and support tasks. Such tasks are mainly addressed to help elderly people in order to possibly prevent and recognize a wide range of common medical problems.

In AAL, personalized products rather than complete out-of-the-box solutions are usually preferred since the possible changes of end-users necessities. With the advancement of technology in the market there are more and more electronic devices which are usually produced by different vendors and are engineered on the base of a wide range of different protocols and technologies. Among them there are, environmental sensors, wearable sensors, miniaturized and integrated in clothes and accessories, telemedicine systems to monitor vital parameters (e.g., blood pressure, body weight, level of oxygen in blood, heartbeat), smart kitchen sensors (e.g., deployed in ovens or other appliances) and multimedia management platforms.

However, even taking into account the most recent researches, beside the quality-of-service and the user experience [46], one of the most relevant aspect is still the design of systems that can be revised in order to fit possible requirements' changes as the users' needs evolve.

There is still the need of reliable methodologies to develop sensor networks, in particular focused at supporting system designers during the development of AAL solutions. This is especially useful in the following scenarios, namely (i) the development of an automated home from scratch and (ii) the on-demand redesign of AAL functionalities. In the first, the designer has to select the sensors which needed to be installed according to functional and non-functional project specifications. In the latter, the designer is asked to change an already existing automated home to meet new requirements.

With the goal of improving the efficiency and to avoiding a waste of resources, known to be a relevant and still open issue in AAL [47], the challenge is to verify in which way the already installed sensors can be exploited in order to fulfill the new specifications.

This research is aimed to introduce a goal-oriented methodology focused to support AAL engineers in order to design and/or redesign AAL environments in scenarios like those previously discussed.

Generally, the methodology is aimed at taking into account requirements elicitation and analysis as part of the professional practice of development in the AAL field, keeping track of the relation between requirements, goals and solutions.

A relevant aspect of the methodology described (and major original contribution) is its goal-oriented approach. Indeed it involves (1) the representation of AAL goals through a formal language and their decomposition in tasks to be fulfilled, and (2) the reference to a logic language aimed at representing the knowledge and the defining a set of reasoning functionalities that provide advanced functionalities helpful at design time.

In order to better explain the proposed methodology, some basic definitions are now introduced. In the field of computer science, ontologies represent an efficient way to formally model the structure of a system and the most significant entities and relations amongst them that comes from observations. In the field of experimental sciences the research is devoted to model and discovery physical phenomena. Unlike that, ontologies focus on the structure / nature of things independently by their physical existence. In computer science, the concept of existence is pragmatic, in fact as stated in [48] "For AI systems, what exists is that which can be represented." In particular it was the year 1993 when Gruber defined an ontology as an "explicit specification of a conceptualization" [48]. A further important contribution comes from the definition that was given by Borst by which an ontology is a "formal specification of a shared conceptualization" [49]. The representation of something in an ontology implies a decision related to design process. With the aim of supporting the design of an ontology, several criteria [48] must be taken into account:

- *clarity:* in fact an ontology has to communicate the meaning of defined terms thus definitions should be objective
- *coherence:* an ontology sanctions inferences which need to be consistent with the given definitions
- *extendibility:* the use of a shared vocabulary must be taken into account during the design process

- *minimal encoding bias:* the conceptualization should be specified at the knowledge level independently by specific symbol-level encoding
- *minimal ontological commitment:* an ontology should make as few claims as possible about the world which is modeled

The development of an ontology is characterized by many benefits if compared to other data models, such as reported in [50]:

- *Knowledge sharing:* the adoption of context ontologies provides computational entities (e.g. agents and services in ubiquitous computing contexts) to share common sets of concepts related to the context while interoperating with each other.
- *Logic inference:* context-aware computing can takes advantage of several existing reasoning mechanisms to infer high-level conceptual context from low-level and raw context. Moreover it can be exploited to check and solve possible inconsistencies of the context knowledge.
- *Knowledge reuse:* reusing already defined ontologies related to different domains can lead to composing a large-scale context ontology without the need of starting from scratch.

The proposed approach takes into account the requirements' elicitation for AAL environments on the basis of the definition of goals, according to well-known standards. Subsequently, it is defined a modular ontology representing goals, tasks, measures and sensors in a formal machine-understandable format. The introduced ontology represents a reference knowledge base for the community of system designers, which can cooperate for its definition and extension. At last, it is introduced a set of logic-based reasoning functionalities in Prolog in order to perform inference and retrieve new knowledge from the ontology.

This chapter is structured as follows: Section 2.2 introduces the proposed methodology. In Section 2.3 it is presented GoAAL, the proposed ontology for goal-oriented development of AAL environments. Section 2.4 reports on reasoning support services for ontology management and for guiding the development process. Such functionalities will be exploited in Section 2.5 to support the development of an AAL environment. In Section 2.6 a brief overview on related work is provided. In Section 2.7 the presented approach is evaluated discussing the obtained results. Finally, Section **??** draws some conclusions.

2.2 Methodology

The goal-oriented methodology introduced in this work is a contribution given by the missing of well-known practices in the development of an Ambient Assisted Living environment. The introduction of a methodology which relies on known engineering standards and technologies is aimed at answering the following questions: (i) the need to control how the design requirements are elicited, evolved and maintained as a part of engineering design practice, and (ii) the need to characterize and communicate about design requirements defined for the project, that are points of interest on one side for designers, on the other side for final customers. The methodology can be divided into two steps: the *elicitation and analysis of the requirements* and the exploitation of an *ontology for AAL* supporting a goal-driven decomposition of requirements to determine which devices are needed to fulfill them.

Living assistance goals are classified corresponding to the degree of emergency of the recognized situation. In detail, we refer to the following categories according to the degree of emergency usually employed in *triage* for estimating the priority of patients' treatments based on urgency conditions:

- red code: critical issues that are harmful and need to be immediately managed by alert propagation and proactive actions that could contribute to solve the state of emergency;
- yellow code: issues that can be potentially dangerous. They can led to harmful consequences thus they must be recognized in short times in order to avoid more severe repercussions;
- green code: actions that can improve general health conditions. They don't need to be managed with urgency.

We also introduce a further typology, namely the white code, comprising all those support services which are not critical to detect issues but can be considered a valuable contribution for home management, energy saving, comfort, and so forth.

In the following, for the analysis of goals we refer to the Goal-oriented Requirement Language (GRL)[51], a language for supporting goal-oriented modeling and reasoning of requirements. It provides constructs for expressing various types of concepts that appear during the requirement process, among which we refer to intentional elements like goal, task, softgoal, and resource, without discussing the operational details of processes or system requirements. This allows to take a higher-level perspective towards modeling the current or the future software system and its embedding environment. The basis for GRL is *i**[52], a framework with an agent-oriented approach for requirement engineering. It is usually adopted in contexts where multiple parties with strategic interests may be reinforcing or conflicting in relation to each other, such as business process redesign, information systems requirements engineering, or the design of agent-based software systems. GRL, together with UCM (Use Case Maps), constitute the User Requirement Notation (URN)[53], that is adopted as international standard since 2008. The decomposition of high-level goals into more detailed subgoals allows to take into account several aspects in the context of ambient assisted living. Finally, (sub)goals are further detailed in tasks, that specify particular ways of meeting the needs stated in the goals. Several types of relationships among intentional elements are available in GRL, among which:

- decomposition, that provides the ability to define which other elements need to be achieved or available in order for a task to perform;
- means-end, to describe how goals are achieved. Each provided task is an alternative means for achieving the goal. Normally, each task may have different types of impacts on softgoals, which serve as criteria for evaluating and choosing among each task alternative;
- contribution, that describes how an intentional element contributes to the satisfaction of another intentional element. Several contributions are possible, according to the degree and polarity of the relation, e.g. some+, make, help with positive meaning and break, some-, hurt with negative sense;
- correlation, that is aimed to express knowledge about interactions between intentional elements in different categories, and to encode such a knowledge. It is similar to a contribution, except that it is not an explicit desire, but is a side-effect.

In Figure 2.1 a fragment of the goal diagram dealing with red code issues is shown. Circles and hexagons represent goals and tasks respectively. The edges linking a goal to a set of subgoals represents the decomposition relation, that describes which elements are needed to satisfy the goal. The arrows labeled "Help" represent the positive yet uncountable contribution of a task to achieve a goal (e.g., an irregular heartbeat or an excessive perspiration are both well-known symptoms of a heart attack and hence positively contribute to its recognition), while the label "Make" means that the task itself is capable to completely satisfy the corresponding goal (e.g., detecting seismic waves is enough to recognize an earthquake).

The goal diagram, together with all knowledge about goals, tasks, sensors and measures collected by sensors are codified in GoAAL, an *ontology* written in a formal, machine-understandable language. After the identification of the goals, users can exploit GoAAL in order to decompose a high-level goal in subgoals and tasks, and then determine which measures and sensors are needed for its fulfillment. Next section is devoted to describe in detail the ontology schema.

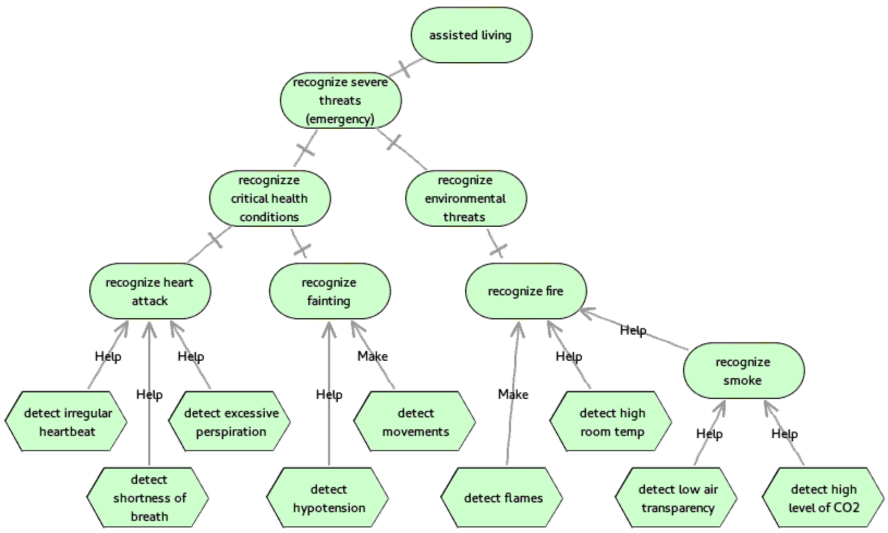

Figure 2.1: A fragment of the goal diagram.

While this last is generic, the definition of ontology instances depends on the target application scenario, and as such it is a required step at the beginning of the development[1]. Finally, as we detail in Section 2.4, through the exploitation of ontological knowledge various types of support functions based on *reasoning* are defined. They are aimed both to retrieve explicit and hidden knowledge in the ontology, and to provide more advanced functionalities for design. For instance, the possibility to determine which is the minimum set of sensors to satisfy a set of requirements or to understand whether an already existing set of sensors can be reused for a scenario with changed requirements.

2.3 An AAL ontology

The main purpose of the GoAAL ontology is supporting a goal-oriented development of an ambient assisted living system that can be considered as a network of different electronic devices. With the aim to drive and give support to the system architecture during the choice of the sensors to install, the ontology focuses on goals of the AAL domain and all the necessary tasks to fulfill them.

In recent times, many methodologies have been proposed for the ontology development (see [55] for a complete survey). By the way, despite the well-

[1]Hereafter in the examples we make reference to specific measures and sensors adopted for the HiCMO project [54].

known reputation of semantic technologies, no widely accepted standard has emerged . In several cases more than one overlap can be recognized among such methodologies: in fact, they are made of interactive and iterative steps that, starting from the identification of main representative terms of the domain, lead to the final ontology through several steps. Some researches [56, 57, 48] provide the theoretical groundwork for knowledge engineering in the context of domain ontologies. The work of Schreiber et al. [56] focuses on corporate knowledge management, for the design and implementation of knowledge-intensive information systems, rather than ontology engineering. Instead, *Enterprise Ontology* [57], which influenced much work in ontology community, represents one of the first attempt to define a sound development methodology. Gruber et al. [48] introduce criteria for ontology design. The formalization guarantees the clarification and disambiguation related to the meaning of terms.

In particular, in this work the adopted approach is based on [58, 59]. It starts from the determination of domain and scope of the ontology, which in this case is the formal representation of goals, tasks, measures, sensors and their mutual relations. Additional steps, that we discuss in the next sections, involve:

- a preliminary analysis for defining the scope and domain of the ontology (Section 2.3.1)
- the reuse of existing ontologies (Section 2.3.2)
- the definition of classes and relations aimed at representing the domain of interest (Section 2.3.3)
- the implementation which is aimed at translating axioms in a machine-readable language (Section 2.3.4)

2.3.1 Domain and scope

The main purpose of the GoAAL ontology is covering the domain of ambient assisted living and smart homes representing the developed goal diagram in a formal way. The ontology will define the relations between tasks and measures and between measures and sensors capable to detect them. Such an ontology will be adopted to provide a goal-driven support tool to those developers involved in the ambient assisted living design. To this end, possible competency question [59] to which ontology will provide answers that comprise these questions:

- which are the goals, softgoals, tasks and resources which are relevant in an ambient assisted living scenario?
- which tasks must be fulfilled in order to satisfy a certain high-level goal?

- which sensors are the most suitable to fulfill a given task? Or a certain high-level goal?

- which is the smallest set of sensors which is able to grant a certain set of measures?

- which way is it possible to reuse a given set of already installed sensors? Can they be exploited to fulfill other achievements?

2.3.2 Reuse of existing ontologies

The development of the GoAAL ontology has been carried on taking into account of other existing ontologies. In particular starting from the analysis of the state of the art of AAL ontologies with the purpose to reuse vocabularies or ontologies which are relevant in the AAL domain. In the following we describe the most important aspects of the external ontologies that we selected for the extension:

- OntoiStar [60] corresponds to the ontological representation of i*, an agent-oriented framework for goal-oriented requirement engineering. It provides constructs for the goal representation and their decomposition in (sub)goals and tasks specifying particular ways of meeting the needs stated in the goals. Softgoals and resources are other concepts introduced in OntoIStar. Moreover, several types of relationships amongst the above mentioned elements are available (e.g., decompositions, contributions).

- DogOnto [61] is an ontology focused on the concept of interoperability amongst the different systems within the same smart environment. It includes device/network independent description of houses, taking into account architectural elements and devices, which are classified as either controllable or non-controllable.

- KPIOnto [62] is aimed at representing indicators (PI) conceived to monitor performances in business contexts. PIs correspond to the notion of measures, and are typically classified as either *atomic* or *compound*, depending on the availability of a mathematical formula stating how they can be calculated from other PIs. Differently from similar ontologies, on the top of KPIOnto a set of logic-based functionalities are defined, capable to manipulate formulas for a variety of reasoning tasks.

- The Semantic Sensor Network (SSN) ontology [63], developed by the W3C Semantic Sensor Networks Incubator Group (SSN-XG), describes capabilities and properties of sensors, the act of sensing and the resulting observations. The ontology is aligned to the DOLCE-Ultralite foundational ontology[2] to further explain concepts and relations.

From now on, prefixes *ois*, *dog*, *kpi* and *ssn* are used in the following to refer respectively to OntoiStar, DogOnto, KPIOnto and SSN ontologies.

2.3.3 GoAAL architecture

This section is devoted to describe the ontology architecture. It's divided into three distinct modules, that will be discussed in the following:

- *The Goal module* describes the goals and their hierarchical decomposition in subgoals and tasks related to an ambient assisted living context in order to understand which domain objects should be measured starting from the high-level goals analysed in the goal diagram.

- *The Measure module* is devoted to characterize measures and their relations with domain objects, in order to understand which measures can be exploited to fulfill a given task.

- *The Sensor module* is focused to the definition of sensors, their properties and their specific functionalities. This module is relevant in order to determine which are the best sensors that can be exploited to obtain a given measure or which are the actuators that are capable to operate on a given object.

The three modules of GoAAL ontology and their interconnections are shown in Figure 2.2. In particular the key concepts are represented within white boxes, the extended classes from other ontologies are represented with coloured boxes. Their mutual relations are shown as well in the picture.

[2]http://www.loa-cnr.it/ontologies/DUL.owl

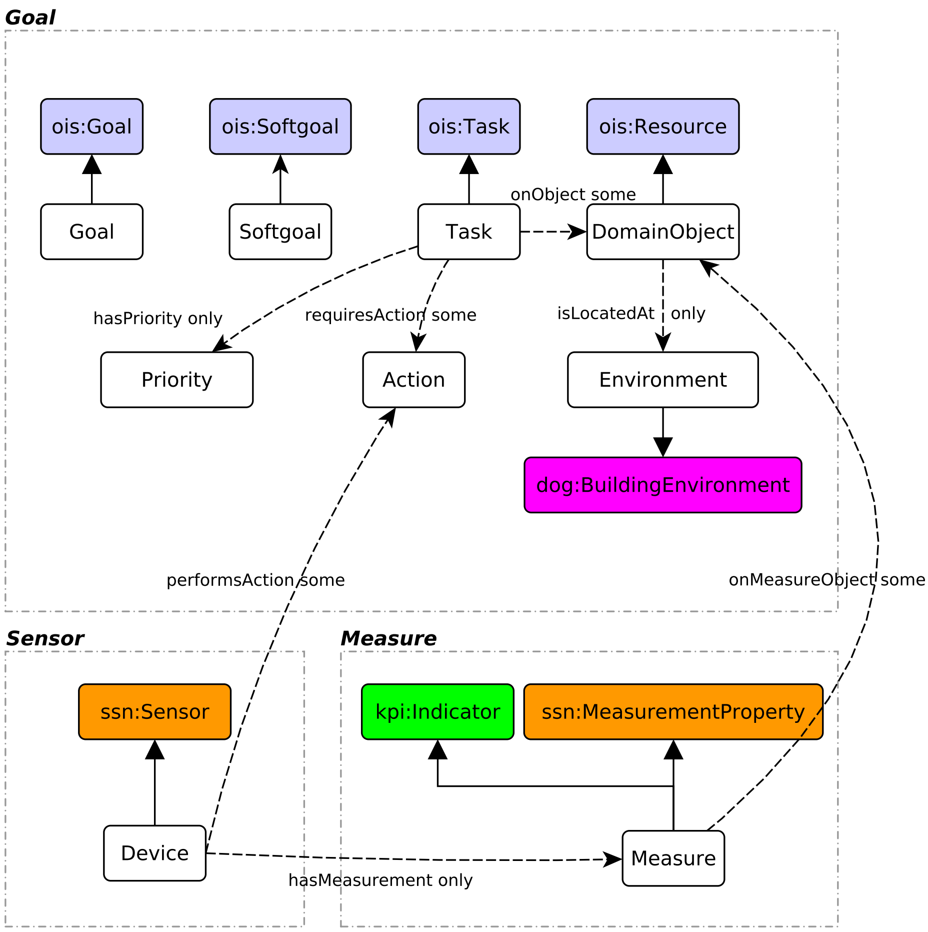

Figure 2.2: The GoAAL ontology schema.

Goal module

This module is aimed at modeling goals, soft-goals, task and resources. This module formalizes the ontology specifications of the Goal-oriented Requirement Language (GRL), and extends it using the notions of Action, Environment and Priority. The Goal module is shown if Figure 2.3 and its classes and relations are explained in detail below:

- **Goal**. This class represents goals ($\sqsubseteq$ `ois:Goal`) and its decomposition in subgoals. In the proposed approach, goals are classified in agreement to the degree of emergency of the recognized situation. Red (critical issues that have to be managed as soon as possible), yellow (risky issues that must be recognized to avoid more serious repercussions) and green code (actions which are capable to improve general health conditions). We also refer to a further typology, named the white code, that covers non-critical services which are conceived to provide support for comfort and home management. As for example, we can just imagine a red code goal related to recognition of environmental issues. Among its subgoals, the *FireDetection* goal can be directly fulfilled through the *FlamesDetection* task to detect the presence of flames within the apartment. The *HighTemperatureLevelDetection* task contributes as well to the goal, as an abnormal high temperature measured within a room may witness the presence of fire. In addition, *SmokeRecognition* subgoal can help to achieve the goal at hand. In turn, this can be achieved through tasks *HighLevelOfCO2Detection* or *LowAirTransparencyDetection*. A subgoal is the recognition of smoke that is achieved through the detection of high level of CO_2 or low transparency of the air.

- **Softgoal**. A softgoal ($\sqsubseteq$`ois:Softgoal`) represents conditions for which there is no clear-cut criteria to judge its satisfaction. It is mainly used for non-functional requirements, e.g. it can be important to assure $Usability$ of devices in contexts with elderly people.

- **Task**. A task ($\sqsubseteq$`ois:Task`) is related to the performing of an action ($\forall$requiresAction.`Action`) over a resource ($\forall$onObject.`DomainObject`). For example, the task $MonitorIrregularHeartbeat$ requires a $Monitoring$ action over $Heartbeat$ domain object. A Task may be associated to a priority ($\forall$hasPriority.`Priority`).

- **Priority**. The priority of a task (e.g., $High-priority$, $Low-priority$).

- **Action**. The type of action that can be executed over a Resource (e.g., $Monitoring$, $Enactment$).

- **DomainObject**. The representation of the domain's objects (e.g., vital parameters like $HeartBeat$, environmental quantities like $Temperature$ or $LevelOfCO2$). It is defined as a subclass of Resource in the OntoiStar ontology ($\sqsubseteq$ `ois:Resource`). Domain objects are hierarchically classified according to their typology. In the AAL field, subclasses comprise relevant parameters related to health conditions (e.g., body temperature, breath, heartbeat) and physical variables related to the environment (e.g., room temperature, humidity, seismic waves). A domain object is also located within a certain environment ($\forall$isLocatedAt.`Environment`).

- **Environment**. This class is used to represent all the possible areas of an apartment. It comes from an extension of the corresponding class of the DogOnto ontology (⊑`dog:BuildingEnvironment`).

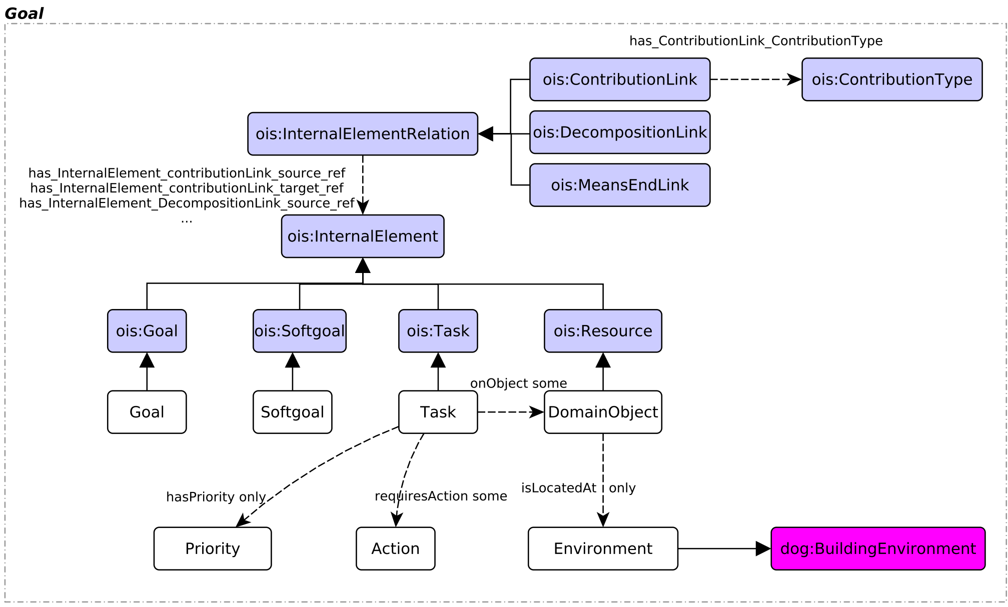

Figure 2.3: Goal module of the ontology.

Measure module

The purpose of this module is formalizing measures gathered by sensors. Measures can be classified on the base to their structure and the availability of an associated formula into the following categories:

- simple measures, corresponding to instances of the class Measure. They can be further detailed into:
  - atomic measures in case that a formula is not provided
  - compound measures in case that it is provided a formula which describes how the measure can be mathematically computed starting from other measures

The class Measure extends a class from the SSN ontology (⊑`ssn:MeasurementProperty`), and also the class Indicator defined in the KPI-Onto (⊑`kpi:Indicator`). This last is described in the following together with relevant classes from the same external ontology:

- **kpi:Indicator** is the most significant class of KPIOnto, and its instances describe any metric that can be monitored, e.g. $HeartBeatRate\_bpm$, $CO2Level\_ppm$. Properties of an indicator are: name, acronym, definition, the aggregation function (∃kpi:hasAggrFunction.`kpi:AggregationFunction`), the unit of measurement (∃kpi:hasUnitOfMeasurement.`kpi:UnitOfMeasurement`) and a formula (∀kpi:hasFormula.`kpi:Formula`).

- **kpi:UnitOfMeasure** represents the symbol and the description of the unit chosen for the indicator, making reference to an external ontology, namely the Measurement Units Ontology[3].

- **kpi:AggregationFunction** is the function through which a measure can be aggregated along some dimensions (e.g., the $MeanDailyTemperature$ is given by averaging temperature values retrieved during the day).

- **kpi:Formula**. It is used to calculate a compound indicator starting from other indicators. Each formula is characterized by its graphical representation, its semantics and the references to its components, which are in turn formulas. For instance, the $BodyMassIndex$ is derived by $Weight * Height$ or the mobility of a user can be evaluated by a smart shoe device as $NumSteps * StepLenght$.

A measure is also related to the specific domain object to which it refers (∃hasObject.`DomainObject`).

[3]http://idi.fundacionctic.org/muo/muo-vocab.html

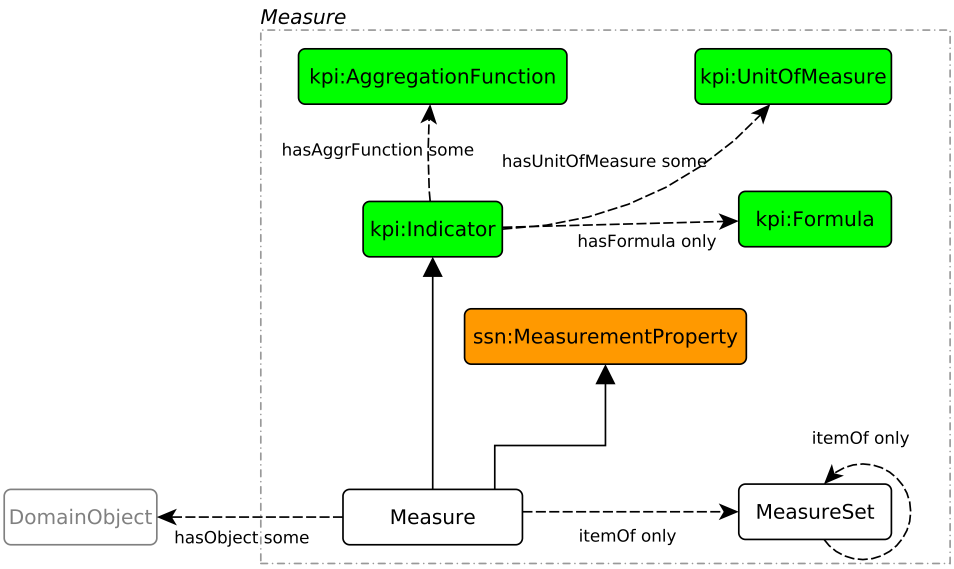

Figure 2.4: Measure module of the ontology.

Sensor module

This module includes a Sensor class which is devoted to describe the full set of electronic devices installed in a smart environment; it extends the class Sensor in the SSN ontology ($\sqsubseteq$`ssn:Sensor`). The module is depicted in Figure 2.5.

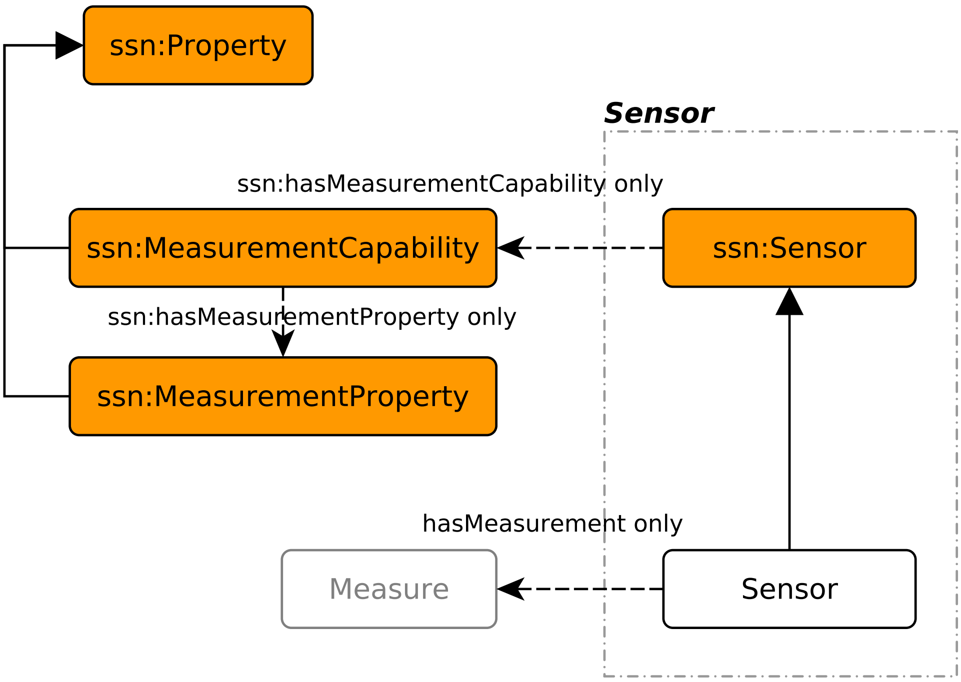

Figure 2.5: Sensor module of the ontology.

- **ssn:Sensor**. Within SSN, a sensor is an entity which has the capability to follow a sensing method and observe some property (∀ssn:observes.`ssn:Properties`) strictly related to real-world phenomena. Sensors can be electronic devices, computational methods or any other thing able to follow a Sensing Method to observe a Property.

- **ssn:MeasurementProperty**. It is an observable quality of an event or object. Two subclasses are: MeasurementProperties which is used to specify technical information about the sensor (e.g. frequency, accuracy, range, precision) and MeasurementCapabilities that collects together measurement properties and the environmental conditions in which those properties hold, representing a specification of a sensor's capability in those conditions. The class Measure extends class Property and this last is related to the measure that is observed. A capability has a measurement property (∀ ssn:hasMeasurementProperty.`MeasurementProperty`), whereas this last is detailed by specifying the value of the property (∀ hasMeasurementPropertyValue.(∃ hasQuantityValue)) and/or, in case the property is associated to a range, the minimum and maximum values (∀ hasMeasurementPropertyMinValue.(∃ hasQuantityValue)).

- **ssn:MeasurementCapability** is a subclass of ssn:Property that collects together measurement properties and the environmental conditions in which those properties hold.A capability has a measurement property (∀ssn:hasMeasurementProperty.`MeasurementProperty`), whereas this last is detailed by specifying the value of the property (∀ssn:hasMeasurement-PropertyValue.(∃ssn:hasQuantityValue)) and/or, in case the property is associated to a range, the minimum and maximum values (∀ssn:hasMeasurementPropertyMinValue.(∃ssn:hasQuantityValue)).

For example, $CO2DetectionSensor_{123}$ is manufactured by "SomeCompany$_1$" and hasMeasurementCapability.frequency of "3/min" and hasMeasurement $CO2Level$.

2.3.4 Implementation

The design process of an ontology is based on formal languages. In particular in 2004 W3C Consortium recommended the Web Ontology Language (OWL) as the main formalism to be adopted for the ontology representation [64]. Thus OWL is widely adopted language in sciences communities, where it has quickly become a de facto standard for ontology design and data interchange. OWL originates as a revision of the DAML+OIL ontology language [65], inheriting from them several features of them. An important aspect is that the consistency of the OWL language with respect to lower level Web standards is guaranteed. OWL is based on the RDF model so it relies on a RDF/XML syntax. The RDFS terms are also integrated within the OWL vocabulary.

There exist three versions, namely sublanguages, of OWL that are succinctly summarized below:

- *OWL Lite:* it was originally developed with the aim of supporting the users that needed a classification hierarchy with simple constraints. For instance, even though it supports cardinality constraints, only cardinality values of 0 or 1 are allowed. Development with OWL Lite is almost as difficult as development with OWL DL, because of that OWL Lite is not so widely adopted [66].

- *OWL DL:* it is much more expressive than OWL Lite since it includes all constructs which are present in OWL Lite, extending them with specific features of Description Logic (e.g. class disjoint, conjunction, disjunction and so forth) which enables automated reasoning functionalities [67]. OWL DL provides on one hand computational completeness and on the other hand an high level of expressiveness.

- *OWL Full:* it is the best choice for users who need the maximum level of expressiveness and the syntactic freedom of RDF without computational guarantees. It is meant to be used in situations where expressiveness is more significant than being able to guarantee the decidability or computational completeness of the language. It is not possible to perform automated reasoning on OWL Full ontologies.

The GoAAL ontology is implemented in OWL2-RL[4] language (based on a subset of OWL-DL), with the exception of the measures' formulas for which OWL2 does not allow a proper representation. Hence, in order to exploit them for reasoning purposes, these mathematical expressions are stored in a repository relying on the W3C Recommendation MathML[5] and OpenMath[6], two XML-based standards frequently used to render equations in web browsers. They provide a language to represent formulas in a machine-readable format with a defined syntax and semantics, which allows to perform automatic manipulation on them. Such languages are capable, respectively, to describe the content/presentation of a generic formula and the semantics of mathematical objects.

In Section 2.5 we discuss some application of the GoAAL ontology related to goal-oriented design and development of an ambient assisted living environment. However, the ontology can also exploited for a number of other possible purposes. As an example, GoAAL can be used in context-aware systems for an automated home, in which a sensor network constantly monitors significant measures which are stored in central a repository. In such a case the ontology represent a useful reference vocabulary to semantically enrich relevant terms of the domain (i.e. sensors, measures, rooms) by mapping specific values to ontological concepts. In the same context but from an analytical perspective, it can also be used in a ontology-based data access (OBDA) approach, to make queries over the repository using ontology terms and exploiting reasoning capabilities. The implementation of such external systems or repositories is obviously decoupled from that the ontology, and can refer to technological solutions ranging from relational databases to linked data.

2.4 Querying and support reasoning functions

We have defined many reasoning services thanks to which it is possible to execute queries over the ontology to retrieve useful information. The ontological model can be queried through Simple Protocol and RDF Query Language (SPARQL) to retrieve both information that is explicitly stored and implicitly

[4] http://www.w3.org/TR/owl-overview/
[5] http://www.w3.org/Math/
[6] http://www.openmath.org/

inferable by a reasoner. For instance, possible queries can help to understand which tasks must be performed to achieve a given high-level goal, which measures are devoted to monitor a certain domain object or which sensors are capable to provide a given measure.

More interesting queries on measures' formulas can be executed, for instance to determine which is the minimal set of sensors providing a given set of measure. However, in this case specific reasoning functions must be defined on the top of the model, by exploiting the formal representation of formulas. Indeed, as proposed by authors of KPIOnto ontology [62], in order to define such functionalities it is needed to represent and manipulate formulas within a reasoning framework. To this end, they refer to Logic Programming (LP) as logic-based language for its capability to manipulate both (properly translated) OWL2 axioms and mathematical equations. Indeed, mathematical formulas are represented as facts by referring to an infix notation like:

- `formula('A','B'+'C').`
- `formula('B','D'+'E').`
- `formula('D','F'/'H').`

In the following, we show some reasoning functions proposed in KPIOnto capable to support generic queries about measures and formulas, that we extended with a set of specific predicates for AAL. We also discuss a set of predicates devoted to the management of consistency in the ontology.

In addition, with the aim to support more advanced reasoning functions among measures and sensors, facts of type `hasMeasure(sensor,measure)` are included in the Prolog knowledge base to state that, in the ontology, a certain *sensor* is linked to the specified *measure* through the hasMeasurementCapability/hasMeasurementProperty relations.

The problem related to the manipulation of mathematical expressions is addressed by specific predicated from PRESS (PRolog Equation Solving System) [68], that can be considered as a formalization of algebra in Logic Programming specifically aimed to solve symbolic, transcendental and non-differential equations. Its code can be represented as axioms of a first order mathematical theory and the running of the program can be regarded as inference in such a theory. More in detail, given that predicates and functions in PRESS represent relationships among expressions of algebra, it can be interpreted as a Meta-Theory of algebra. The predicates in which it is organised can manipulate an equation to achieve a specific syntactic effect (e.g. to reduce the occurrences of a given variable in an equation) through a set of rewriting rules. XSB[7] was chosen as logic programming database system for its efficiency.

[7] http://xsb.sourceforge.net/

2.4.1 Reasoning to support development

A set of *basic functionalities* are aimed to enable manipulation of mathematical formulas by extending PRESS (PRolog Equation Solving System)[8], a formalization of algebra in Prolog that includes predicates for various mathematical tasks, like basic symbolic operations, equation solving and equation rewriting.

More *advanced functionalities* are focused on the extraction of "common measures": given a set of measures $\phi = \{m_1, m_2, ..., m_n\}$, common measures of ϕ is the minimal set of measures needed to compute all formulas of ϕ. In [62] such a concepts is implemented through the predicate `indToMea(`L_{in}`, `L_{out}`)`, which takes as input a list L_{in} of measures and generates its common measure set in L_{out}. To give an example, let us consider the following formulas for A, B and D: `formula('A','B'+'C')`, `formula('B','D'+'E')`, and `formula('D','F'/'G')`. If $\phi = \{A,\ B,\ C,\ D,\ E,\ F,\ G\}$, then a solution for `indToMea(`ϕ`,`L_{out}`)` is $L_{out} = \{C, D, E, G\}$. In fact, with D and G we can compute F by reverting the third formula. Then, having D and E, B is obtained according to the second formula. Finally, using the first formula A can be derived from B and C. Let us note that other possible minimal solutions exists, like $\{A, B, D, G\}$ and $\{B, C, F, G\}$. The predicate `meaToInd(`L_{in}`,`L_{out}`)` implements the inverse of indToMea: given a set of available measures L_{in}, the predicate returns in L_{out} all those measures that are derivable from them, e.g. if $L_{in} = \{B, C, E\}$ it is possible to derive $L_{out} = \{A, B, C, D, E\}$.

We extended this set of predicates by defining the following ones, to specifically support the retrieval of information in GoAAL:

- `meaToSen(`L_m`,`L_s`)`: given a set L_m of measures, returns the sets L_s of sensors needed to compute them. This relies on determining from `indToMea` the minimum set of measures actually needed, and then on the execution of a Prolog goal in the form `hasMeasure(X,measure)` for each measure. For example, let us consider a knowledge base with sensors $s1$, $s2$ and $s3$ and measures A, B and C:
  - `hasMeasure('s1','A'). hasMeasure('s1','C').`
  - `hasMeasure('s2','B').`
  - `hasMeasure('s3','B'). hasMeasure('s3','C').`

  In such a case, given L_m=$\{A,\ B,\ C\}$, then $L_s = \{s1, s2, s3\}, \{s1, s2\}, \{s1, s3\}$.
- `senToMea(`L_s`,`L_m`)` given a set L_s of sensors, returns the set L_m of measures that can be calculated: at first the list of available measures is retrieved by exploiting the goal `hasMeasure(sensor,X)` for each sensor in L_s. Then, `meaToInd` is executed to expand the list of obtained computable measures.

[8]http://dream.inf.ed.ac.uk/software/press/

- `taskToSen(`L_t`,`L_s`)` given a set L_t of tasks, returns all the sensors L_s capable to achieve them. At first the set of domain objects related to the tasks are retrieved from the ontology, and in the same way all the measures linked to such objects are found. Then, predicate `meaToSen` is executed to retrieve the sensors.

2.4.2 Consistency management

In the context of this work, the ontology is meant to be populated before the beginning of the development phase. Usually this happens in an incremental way, as developers reuse already existing definitions of sensors and measures for following projects. A typical problem in collaborative and incremental knowledge bases is consistency management. This can be usually achieved by evaluating the (in)coherence of a new instance with respect to the ontology schema, by exploiting a consistency service available in OWL reasoners.

For what concerns formulas, a check is needed to verify whether a new added formula is consistent with the already existing formulas, from a logical-mathematical point of view. On the contrary, the answer to a query could produce inconsistent results. In this case the consistency check is implemented as a Prolog predicate according to the following definition.

Let $\mathcal{KB}$ be a knowledge base containing formulas. Given a new measure m_i, its formula f_i is consistent with $\mathcal{KB}$ iif (a) f_i is unique in the ontology (there is no identical formula), (b) its formula is not equivalent to a formula already in $\mathcal{KB}$ and (c) its formula must not contradict any other already defined formula.

As for the first condition (a), the `identical(Equation,X, X=Solution)` predicate is introduced to avoid to insert twice the same definition in the ontology. The reasoner firstly rewrites the `Equation` so that the variable `X` is only on the left side. Then it searches the whole theory for a formula having the same structure of the rewritten `Equation`. Moreover, (b) during ontology population, it is useful to individuate and manage duplications, i.e. given two formulas f_1 and f_2, they are equivalent if f_1 can be rewritten as f_2. The predicate `equivalence(Equation,X,Y)` implements this behaviour by relying on functionalities for formula manipulation, to find formulas `Y` that are equivalent to `X=Equation`. Finally, (c) we defined a predicate `incoherence(Equation,X,Y=S)` useful to verify that when a formula `X=Equation` is added or existing formulas are updated or deleted, the change do not contradict the ontology. A formula f_2 for a new measure m_2 is coherent with the ontology if, whenever there exists a formula f_1 for a measure m_1 already defined in the KPIOnto, such that f_2 can be rewritten as f_1, then the equivalence $m_1 \equiv m_2$ does not contradict the ontology. Otherwise we have an incoherence.

2.5 A GoAAL Application

As previously discussed in Section 2.3.4, GoAAL is developed with the aim of being a useful support for developers in the design of ambient assisted living environments. The previous Section showed how low-level competency questions can be answered by exploiting SPARQL queries and reasoning capabilities. On the other hand, this Section reports about possible usages of the ontology for a number of potential real-world applications. In particular we focus on the following case studies:

1. the development of a cockpit devoted to continuously monitor temperatures within various rooms of a house. In such a scenario the major challenge is overcoming syntactic and semantic heterogeneities of sensors that come from different producers / manufacturers.

2. the on-demand redesign of AAL functionalities. In this situation the developer is asked to change an already existing automated home in order to fit new requirements and specifications (e.g., the patient is under treatment for a disease). With the aim to optimize costs and to avoid waste resources, the main purpose is to verify whether the already installed sensors are able to fulfill the new requirements.

3. sensor matching based on design requirement. In this scenario the automated home is developed from scratch. Hence, the developer must choose which sensors are to be installed according to functional and non-functional requirements.

The case studies are discussed in the context of a collaborative project called HicMO, that lasted from 2011 to 2015 and involved thirteen Italian companies specialized in microelectronic devices, house appliances and telecommunication systems [54]. The high-level goal of HiCMO was to build a platform providing advanced services which are focused on monitoring domestic activities in order to ensure the wellness of elderly people. Many kind of devices are taken into account:

- environmental sensors: they monitor temperature, humidity, pressure, power consumption, air and lighting quality of the various zones of the house.

- wearable sensors: they are miniaturized and integrated in clothes and accessories. The e-Moves sensor is capable to acquire information of user's gait and walk activity that can be relevant to analyze the health status of the user [69, 70]. There are also sensors which are integrated on a t-shirt and are aimed at revealing information about the user's stress level. The user is also provided with a special ring with an RFID tag that is used to monitor the presence of the user in specific areas of the house.

- telemedicine system: it is composed by several devices and is used to monitor some vital parameters of the users like blood pressure, body weight, level of oxygen in blood and heart rate.

- smart kitchen: it integrates various sensors to reveal significant events related to the user activities. For example the fridge communicates to the system when a special medicine box is open by the user. Drawers and kitchen cabinets can be automatically locked if the system detects the presence of children or unauthorized people in the room.

- multimedia management system: it is composed of a SmartTV adapter and an e-Learning platform. The first turns any TV into a device able to play multimedia contents, while the second is used to help users by training.

Central to the project are the concepts of compatibility and interoperability of heterogeneous devices, coming from different vendors / manufacturers and thus using different technologies and protocols. The interoperability is achieved by representing each device as a software module called Smart Object (SO) which is devoted to represent its functionalities and capabilities. This way, we can consider each SO as a service able to transmit information over HTTP to the Domotic Integration Platform. Such a platform is constituted by:

- *hardware part* which is a centralized gateway that allows physical and functional integration amongst the different electronic devices involved in the project.

- *software part* which implements the reference model used to formalize the data model adopted for the Smart Objects description. Each SO is described through specific XML descriptors that define its functionalities, interfaces and gathered measurements.

2.5.1 Development of a cockpit for monitoring home temperature

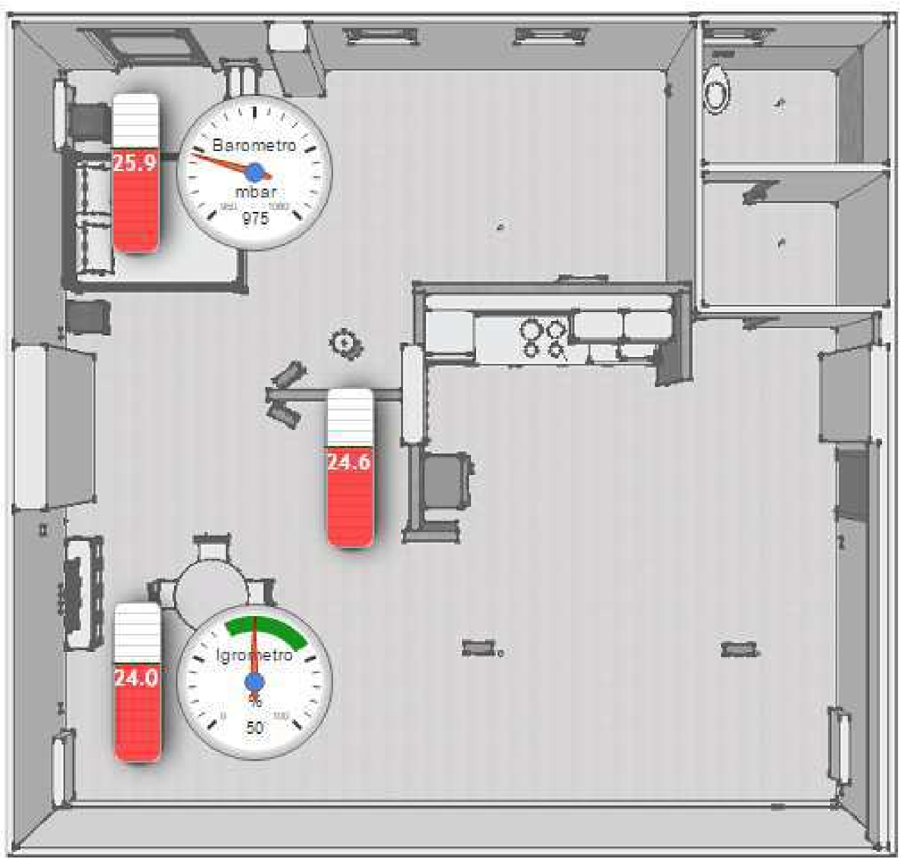

Figure 2.6: HicMO: prototype of the cockpit for monitoring room temperature

In this scenario the user is interested in a cockpit through which she can monitor temperature in different rooms of the house. Let us suppose that a further specification refers to a sampling frequency greater than 10 times per minute. At first the developer selects from Domain Object class the corresponding instance and retrieves the related measure, i.e. temp_room. Let us suppose that for the HicMO project the following environmental sensors for measuring room temperature are defined in the ontology:

- s1 (unit of measure: C°, frequency: 20/minute);
- s2 (unit of measure: C°, frequency: 6/minute);

- s3 (unit of measure: C°, frequency: 12/minute);
- s4 (unit of measure: F°, frequency: 15/minute)

Within the set of sensors only s1, s3 and s4 satisfy the non-functional requirement related to the working frequency. However, they yield as output a value in different units of measurement, namely Celsius degrees for s1 and s3, and Fahreneheit for s4. Apart from that, s1 is also a barometer to measure air pressure, while s4 embeds also an hygrometer capable to measure the moisture content of the air. The developer thus decides to include also these values in the cockpit to enrich the information provided to the user, as in Figure 2.6 where the prototype used in the project is shown.

2.5.2 On-demand redesign of AAL functionalities

In this scenario let us assume a developer is asked to redesign the set of functionalities already offered by a domotic system in the HicMO project according to a new goal. By following the decomposition of such goals, eventually a set of tasks to fulfill is retrieved. From them, the set of requested domain objects and consequently the related measured M_r are defined. Let us define the set of sensors already available by S. By execution of predicate `senToMea(S,M)`, introduced in subsection 2.4.1, the set M of measures provided by S is retrieved. At this point, if the new requested measures are included in M (i.e., $M_r \subset M$), it means that they are already available. In this case the developer can just take into account the implemented system and reuse or relocate the sensors according to the new specifications. On the contrary, if $M_r \not\subset M$ it means that there are some requested measures in M_r that are neither available nor can be computed from those available. Let us then call such a set $M_r^{'} = M_r \setminus (M \cap M_r)$.

In order to find the (minimal) set of sensors needed to compute $M_r^{'}$ it is possible to refer to the previously discussed predicate `meaToSen`.

To make an example, let us suppose to have the three sensors in three different rooms, that are focused on measuring B. Each sensor is actually capable to detect a larger set of measures as follows:

- sensor s1 in room 1 measures A and B
- sensor s2 in room 2 measures B and C
- sensor s3 in room 3 measures B,C and D

The changes in the environment are such that now also measure Z must be added to the required measures set, i.e. $M_r = \{B, Z\}$. Such new measure is not directly provided by anyone. However, by knowing that $Z = B + C$, the system can derive that $s1$ and $s2$ are enough to calculate Z from its operands. As for

$s1$, instead, C is missing and hence Z cannot be computed. Note that being in different rooms, in general, the evaluation has to be made for each room separately: hence, we can note combine B provided by $s1$ with C provided by $s2$ or $s3$. As a consequence, a new sensor must be added to calculate Z also in the room 1. According to the described functionalities, the system suggests the designer with two possibilities, either measuring C or directly measuring Z, and provides as output all sensors that are capable to measure one of the two (that are also compatible with other sensors and with the room at hand, and satisfying other possible constraints, like costs, vendors, I/O format).

2.5.3 Goal-driven requirement for development

In a more general situation, the automated home is developed in a goal-driven fashion, starting from high-level goals that are then decomposed in subgoals and eventually in tasks. The classification of abstract goals is done according to the severity of the corresponding situation to be managed, i.e. red, yellow and green codes. Once the tasks have been identified, the corresponding domain objects are retrieved. Such objects can be measured, in general, by several alternative measures, each of which can be provided by different sensors. For such a reason, the predicate `taskToSen` is called. To make an example, let us assume that a designer is asked to recognize environmental threats within a red code scenario. In particular let us focus on the detection of fire, that can be simply fulfilled by detecting the presence of flames in the house, as depicted in Figure 2.1. Other tasks can contribute as well, i.e. the detection of an overly high temperature in the room is frequently an evidence of the presence of fire. Finally, also the recognition of smoke (which is a subgoal) can help, and this last is achieved through detection of high level of CO_2 or low transparency of the air.

After having identified the set of tasks, each of which will contribute differently for the goal at hand, a set of corresponding domain objects can be found via SPARQL queries, namely $Flames$, $RoomTemperature$, $Smoke$, $LevelOfCO2$ and $AirTransparency$. For each of them, a query allows to retrieve the set of related measures, e.g., $RoomTemperatureValue$ for room temperature. Measures can then be reduced through the execution of indToMea predicate, in order to find the minimal set actually needed to satisfy the requirements. Finally, the corresponding sensors are found. In case further non-functional requirements should hold, for instance related to the maximum budget at disposal, or to technical specification of sensors, they can be used to filter the results, or to rank them according to some quantitative criteria.

2.6 Related Work

This section is devoted to discuss about the most relevant contributions related to this work that can be found in the Literature. They range from AAL platforms to sensor networks ontologies and methodologies for goal modeling. In fact, during the last years, much effort has been put in the definition of knowledge models in order to represent relevant concepts in the field of Ambient Assisted Living (AAL), for a variety of support tasks ranging from integration of heterogeneous devices to monitoring/analysis of sensor networks.

2.6.1 Ambient Assisted Living platforms

In the last decade, much energy have been spent by the research community in order to provide novel and cost-effective solutions in the area of Ambient Assisted Living.

It is in the year 2007, when J. Nehmer supported by his research team. [71] introduces an integrated system concept in the Living Automation field. Their system is principally targeted to support elderly people and persons with physical impairment in their own homes. Their approach relies on Ambient Intelligence (AmI) techniques, in particular they introduce a classification of the needed services according to their degree of relevance. The home assistance services are classified in three major categories, in fact the researchers propose the adoption of emergency treatment services, autonomy enhancement services and comfort services.

A further interesting approach is related to the SOPRANO project [72, 73] introduced by Klein et al. in 2008. Their solution, similarly to the previous one, is focused on providing assistance and support to elderly people, allowing them to live safely and independently within an familiar environment. Such a task is fulfilled thanks to the adoption of smart devices. In particular, the core part of their system, is represented by an Ambient Middleware (SAM) which receives inputs from sensors and user's command to drive the electronic devices which the smart environment is equipped with. The authors introduce an approach relying on the definition of an ontology aimed to make the whole system scalable and reliable. The ontology is structured in low and high level ontologies. The lowest level define a vocabulary for the semantic description of services and the state of all supported electronic devices. The high level ontology provides a vocabulary which is devoted on describing the surrounding environment. Another notable approach is presented in [74]. In this work the researchers address the problem of the behavior recognition of a patient affected by Alzheimer during her daily life. This is done in order to give her proper and needed assistence. The authors introduce a formal activity recognition framework that relies on description logic and possibilistic theory like Markovian models and Bayesian networks. In the healthcare field, another interesting work is presented by Tamura et al. [75]. They propose a smart house designed with the aim of monitoring physiological parameters and vital signs of patients. The Takaoka Techno House is a real smart space equipped with several systems capable to acquire data. There are monitoring systems installed, for instance Electrocardiogram (ECG) measurements are monitored in the bed and the bathtub and body weight is measured on the toilet. The ECG recording from the bed. ECGs are taken during sleep via conductive textiles located in the pillow. Data collected from each monitoring system is stored on the hard drive of a computer within the home. The researchers state that more efficient techniques are needed in order to identify and distinguish inhabitants from each other.

2.6.2 Goal modeling

Nowaday, the concept of *goal* is normally employed in different domains within the computer science field. Recently the researchers introduced a wide variety of interesting approaches related to the goal modeling field, with particular attention to the field of Requirements Engineering field. In this survey [76] the authors highlight how these techniques allow to representation of stakeholder needs and their specification in terms of relationships to other goals or tasks. These relations, on the base of the chosen language, may either be traditional AND/OR decompositions or more qualitative and partial relations of contribution or dependency.

Amongst various methodologies aimed at modeling goals, KAOS (Knowledge Acquisition on autOmated Specification [77]) introduces a formal goal framework which include AND and OR decompositions between goals describing desired states over entities, achieved by actions assigned to agents. KAOS is composed by three main components: (i) a conceptual model for acquiring requirements models; (ii) a set of strategies which are aimed at elaborating requirements models within the presented framework; (iii) an automated assistant representing a reliable support during the acquisition process.

The Annotated Goal-Oriented Requirements Analysis (AGORA) [78] framework is aimed at dealing with goal priorities and methods for solving goal conflicts. Furthermore it provides a way to the selection of alternatives and the measurement of the models' quality. In addition, AGORA is also able to fix possible conflicts amongst different goals and to analyze the impact of requirements changes. The NFR (Non-Functional Requirement) modeling [79] is an approach, introduced by Mylopoulos et al., which is focused to represent human intentions in technical systems. The framework takes advantage of concepts like goals, softgoals, AND and OR decompositions and contribution links, representing potentially partial negative and positive contributions to and from such goals. We conclude this section discussing another relevant approach, the i* Framework [52]. It exploits notations of the NFR Framework, that has been mentioned before. The i* Framework has been incorporated into the Tropos Framework [80], where models are acquired as instances of a conceptual metamodel relying on a set of interdependent actors having their own specific interests, namely goals, which are then analyzed iteratively in order to identify the more specific sub-goals needed for satisfying upper-level goals.

2.6.3 Sensor ontologies

Much work has been done, in the domain of sensor network ontologies. This Section is aimed at reviewing contributions in the Literature that are the most representative to our scenario. SOUPA [81] is an ontology developed with the purpose of modelling and supporting pervasive computing applications. This ontology is expressed trough the OWL language and comprises modular components devoted to represent component vocabularies. These last elements represent intelligent agents with associated places, persons and intentions aimed to support applications in the field of pervasive and ubiquitous computing within a smart environment. The researchers introduce a model which relies on a context broker architecture (CoBrA) where computer entities share the context knowledge exploiting the SOUPA ontologies.

Sommaruga et al., introduce the DomoML ontology [82] whose purpose is defining a standard language which allows the integration of products and services put on the market by different manufacturers and vendors. Authors propose an ontology capable of characterizing a smart home. The starting point is an extension of the former EHS Taxonomy designed by the European Home System consortium. In DomoML there is an higher level of detail in the representation of the electronic devices used in the smart environment.

Other relevant progresses are due to the work of Corno et al.[61]. In fact the authors developed the DogOnt ontology, which is properly designed to extend and improve the DomoML functionalities. An important difference is that every object is classified on the base to its controllability. Since that, focusing on such an approach, the ontology defines for each controllable device a state and one or more functionalities / capabilities.

A significant contribution in the field of sensor networks field is given by the W3C Semantic Sensor Networks Incubator group. In their work [63], they focus on the relevance of semantic technologies in order to deal with data coming from a huge number of heterogeneous electronic decvices. Their OWL ontology enables the representation of sensors, sensor observations, methods used for sensing and the knowledge of the surrounding context.

There are also some ontologies that, in order to be more scalable, have been designed with a layered architecture. In [83] a layered OWL ontology which is named CONON (CONtext ONtology) is introduced. Its upper layer defines general features characterizing the context and is based on the description of general objects and concepts. Furthermore, lowest layers are devoted to describe more specific domains. The authors also introduce a reasoning framework aimed to check the consistency of the context information.

Zhang et al. [84] present a five layered model named Context Stack. Their ontology is principally aimed at describing context-aware systems. In this case, the lowest layer are devoted to collect raw data from sensors, whereas the upper levels are aimed at creating a basis of the context model which guarantees interoperability and integration amongst different and heterogeneous sensors. An additional contribution of the authors is represented by the definition of a Web Based Context Model Ontology that is aimed to support reasoning processes and semantic context sharing.

Another ontology aimed at describing smart objects are presented in the work presented by Meroni et al. [85]. Similarly to this work their ontology is built focusing on the reuse principle. In fact they extend the FIESTA-IoT ontology [86], which is based on the W3C SSN ontology [63], M3-Lite [87] and Q4 [9]. The extension of the FIESTA-IoT ontology includes new concepts like smart object which is not clearly defined in other mentioned ontologies. However a difference with this work, is that in their approach they don't have defined a reasoning framework able to deal with smart object measures.

2.6.4 Formalization of measures

At the best of our knowledge, there is no prior work in the AAL field on the exploitation of formal reasoning on structure of mathematical formulas. Measure modeling and analysis have been however widely studied in the field of Performance Management in enterprises, for monitoring of business activity, and from a technical perspective in Data Warehouse management, where they are sometimes called *indicators*.

Within this last field, indicators are usually defined in a multidimensional context, as the corresponding measures are evaluated along multiple dimensions. However, only few proposals attemps to explicitly take into account some notion of formula (e.g., [88, 89, 90]), and almost always formula representation does not rely on logic-based languages, and hence no inference mechanism and formula manipulation is enabled. For instance, the Metric Ontology [91] is devoted to define metrics together with computation expressions expressed in Operational Conceptual Modelling Language (OCML), that cannot however be

[9] http://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu

formally manipulated.

Formal models are introduced in [92], where authors introduce a goal-oriented meta-modeling framework where goals are defined by means of organizational performance indicators, both represented by referring to a logical language. The approach provides means to evaluate the consistency of goal structures both when designing a new organization and for the specification of an existing organization, also enabling reuse, exchange and alignment of knowledge and activities between organizations like in supply chains. Some predicates expressing relations between performance indicators are defined on an axiomatic basis, however no compositional semantics is represented in this way.

Finally, another approach similar to the one followed in this work is [93], where the Goal-oriented Requirement Language is enriched with the concept of indicators. An algorithm uses values of the source indicators as inputs for the formula and calculates the target value. The work is concerned with languages and methods for specific requirements elicitation and does not deal with general reference models and formula representation does not rely on logic-based languages, hence reasoning is based on ad-hoc modules for direct formula evaluation.

2.7 Evaluation and discussion

The GoAAL ontology describes goals, tasks, objects, measures and sensors related to an Ambient Assisted Living scenario. For what concerns the evaluation of the ontology, it can be done from both the point of view of the capability to describe the domain at hand, and according to an applicative perspective, with respect to the goals it is aimed at. As regards the definition of goals we refers to the ontology that has been developed within the Requirement Engineering community following the i* model [60], while no other work is available in the Literature, at the best of our knowledge.

According to the quality criteria that are taken into account in the ontology engineering methodology, we can discuss the most interesting elements of GoAAL. The choice to reuse terms from external ontologies allows to minimize the ontological commitment, avoiding to define non-essential terms and de-facto making the ontology a wide, modular and integrated knowledge base. The methodology that has been followed, then, guarantees the satisfaction of both clarity and minimization of encoding bias, because of the clear separation between the conceptual representation, with definitions in natural languages, and the translation in a logic language. As regards coherence, the ontology has been checked through Pellet in order to verify the lack of logic inconsistencies. Finally, this ontology can be extended by means of other ontological modules that could be defined in future, as needed, to enhance even more the expressive capabilities of the platform.

For what concerns the applicative perspective, the most computationally complex functionality is related to the execution of advanced Prolog predicates. The core of such functions have been already evaluated in a previous work of ours [62], even if in a different domain (monitoring of KPIs in Virtual Enterprises). Experiments were aimed to evaluate efficiency and effectiveness of the basic predicates for formula manipulation and also the more complex indToMea and meaToInd functions. Given that the new functionalities shown here are given by the composition of such functionalities, we refer the interested reader to the analysis available in the work.

Chapter 3

Mining AAL: discovering users' behaviors

3.1 Introduction

In recent years, much effort was put in the design and development of *smart environments*, that are aimed at improving the life quality of people, providing users with advanced services supporting them during their daily activities [94]. This term, sometimes related to the pervasive/ubiquitous computing paradigm [95], refers to an environment equipped with a number sensors and actuators of different types, including monitoring sensors, RFID tags, power-line controllers, for a variety of different applications ranging from health monitoring to security, light management and energy saving. As such, sensor data represent the log of the activities performed by a user and, in turn, each task fulfilled by a user in the environment can be considered as a sequence of activities. For instance, when a user performs a cooking task in a house, she moves towards the light switch to turn it on, opens the fridge and after a while also switches on the oven to warm the food, and so on until the user leaves the kitchen. Each of these activities triggers several sensors (e.g. doors, motion sensors) and a sensor activation sequence can be seen as the execution, or a trace, of an (implicit) process.

The set of sensors activations (hereafter we refer to each activation as a *micro activity*) may be useful to recognize actual performed activities and, more in general, to derive typical patterns or processes capable to describe the user behavior in performing some task, or *macro activity*, such as cooking.

Analysing these data and deriving a workflow/process model from them is hence helpful not only to gain a better insight on how a certain task is performed, but also to support novel smart services. Among them, for instance, adaptive environments capable to align parameters to the specific behavior of users. These models can also be used to support the identification of anomalous behaviors, that are not aligned with the model generated from sensor data related to past process instances. According to the specific environment, their recognition can be indeed useful to react more quickly to emergencies due to accidents or health issues in a house or a factory, or possible threats in a public area.

Due to the huge amount of data that can be potentially generated, such analyses must be however supported by techniques capable to provide a summarization of event logs monitored by sensors.

In order to introduce the proposed methodology we are giving in the following some basic definitions of process and Process Mining.

There is not exist a single, unique definition of *process*, however an intuitive way to introduce such a concept, is considering a process as a sequence of activities executed in a specific order (it is assumed that there is information about the order in which the events of a case are executed) with the aim to fulfill a given goal. Processes play a significant role in industries, organizations and also in people's daily life. We can refer as example to management of packages in a delivery company, an assembly line in whatever industry or just easily doing some shopping at the market. These activities can be all seen as processes. Process Mining can be considered as a multi-field discipline which sits among machine learning, data mining, and process modeling. The main idea behind process mining is to discover, monitor and also optimize real processes taking advantage of knowledge which comes from event logs available in modern IT architectures [96]. An Event log is an unbiased footprint which represents the process itself. Each process mining technique, ranging from the discovery to the model optimization, relies on event logs.

Process mining techniques are usually classified into three major classes according to their purpose:

- *discovery:* these technique takes an event log, namely a collection of traces, as input in order to give as output a model just taking into account the knowledge extracted from the log. Discovery is the oldest process mining branch, and it is the one with the widest variety of proposed approaches. As event logs grow, process mining algorithms must improve in order to be more efficient scalable. Some of the most known discovery algorithms are: alpha-algorithm [97] which is based on detecting relations among activities of the log, infrequent Inductive Miner [98] or Genet [99] and ILP Miner [100] that are based on the theory of regions, and several other approaches.

- *conformance:* in contrast to discovery techniques, conformance needs two distinct inputs: an event log and a model. Conformance techniques are devoted to check if the behavior modeled represents with precision and accuracy the behavior represented in the log. There are many kind of conformance techniques, see [101] for more information about them.

- *enhancement:* such class of techniques, like conformance, considers both an event log and a model as inputs. By the way, enhancement techniques have the purpose of improving the a-priori model with information contained within the log. This improvement could be done by means of repairing the model to better conform the observed behavior [102]. Another option is to extend the model with additional information.

In the Literature, Process Discovery techniques and algorithms have been used to derive a process model representing a set of given traces, to discover *behavior models* that represent the "standard" behavior of the monitored user in the form of a process model, where activities are tasks or macro-activities, e.g. cooking, reading, watching TV for a smart home. This type of models can be effectively used to get a better understanding of how a user behaves in a given environment. On the other hand, a much less studied yet related task is the discovery of a *macro activity model* from event logs, i.e. a process model representing the flow of sensors activations when a given macro activity is performed. This kind of models serves a similar purpose, but at different level of granularity. In particular, the characterization of a macro activity in this way can be more effective to understand more deeply how a given activity is inherently performed.

With such an aim, we propose a methodology for user behavior analysis that exploits Process Discovery techniques to derive both micro and macro activity models from sensor activation logs in a smart environment. Unlike other work in the Literature, the purpose of this approach is therefore not to generate a predictive model to foresee future states of the process, but to derive descriptive models capable to represent user behavior. The followed approach includes these steps:

- event logs generated by sensor data are characterized in terms of macro activities, annotated by the user, and micro activities;
- logs are preprocessed in order to remove noise and make them compliant with the adopted techniques;
- Process Mining techniques, specifically infrequent inductive mining, are applied over the event logs to derive process models for both micro and macro activity flows.

The main contribution of this research consists in the adoption of Process Mining techniques to represent daily living activities (i.e., macro activities) as a process where each activity is a sensor activation (i.e., micro activities). While the application of Process Mining to derive an activity models is reported by some work in the Literature, to the best of our knowledge this is the first work to propose the derivation of a model for macro activities from logs related to sensor activation. By helping to better and more deeply understand the intrinsic workflows behind a given macro activity, e.g. having a shower, in terms of activated sensors, this can provide useful insights to recognize trends and/or deviances from the model. As a further contribution, in order to assess the effectiveness of the proposed approach, we include an experimentation conducted on a real-world case study from the CASAS project of the Washington State University, specifically dealing with ambient assisted environments. In particular, in the project the behavior of a user and her pet have been monitored for almost two months through multiple sensors deployed in a smart home. The retrieved data have been used in this work as a basis in the application of the approach for the generation of micro and macro activity models.

The rest of the chapter is structured as follows. Next Section is devoted to introduce the case study, while in Section 3.3 we provide some preliminaries on Process Discovery and conformance check. Section 3.4 includes a detailed description of the proposed methodology and presents experimental results. Relevant related work are discussed in Section 3.5. Finally, Section **??** draws some conclusions and presents possible future works.

3.2 Case Study

This section is devoted to describe a case study that we will use as example for referring to within the whole chapter. In particular, we will refer to the *Milan* dataset, which is part of data collected by the CASAS project of the Washington State University[1]. The selected dataset describes the behavior of a user that live with her pet for 58 days in a house, which has been conveniently equipped with some different types of sensors (Figure 3.1 shows the map of the house and sensors positioning), as follows:

- **Motion sensors**, based on infrared technology, they track the user moving in the various rooms of the house. These sensors should be placed at least one per room, in order to understand the user movements during her daily life activities. In Figure 3.1, these sensors are denoted with the prefix M followed by an identifying number (e.g., M002); The motion sensors used in such experiments were of two different kind of typology: spot sensors and wide range sensors. Spot sensors are able to detect motion within a small area and are denoted in the map with a small red circle (e.g. M004 or M005). Wide range sensors are able, in contrast to the previous type, to detect a motion within bigger areas which are highlighted as red shaded zones (e.g. M024 or M027). Motion sensors check the presence of movements at a predefined frequency: if no movements are recognized within the detection range no events are recorded, while if a movement is detected it triggers to ON; in the last case at next sampling time the sensors returns OFF if no other movements are detected. This behaviour can produce noise when either the user stops within the detection area or she moves very fast (see Section 3.4.1 for a more detailed discussion).

- **Open/close sensors** that are able to detect the status of a door. They are installed on the external doors of the house, allowing us to know when someone is leaving or coming back home. These kinds of sensors are denoted with the letter D (e.g., D003);

- **Environmental sensors**. the house is also equipped with some temperature sensors, which are represented with the prefix T (e.g., T002).

[1] http://ailab.wsu.edu/casas/datasets.html

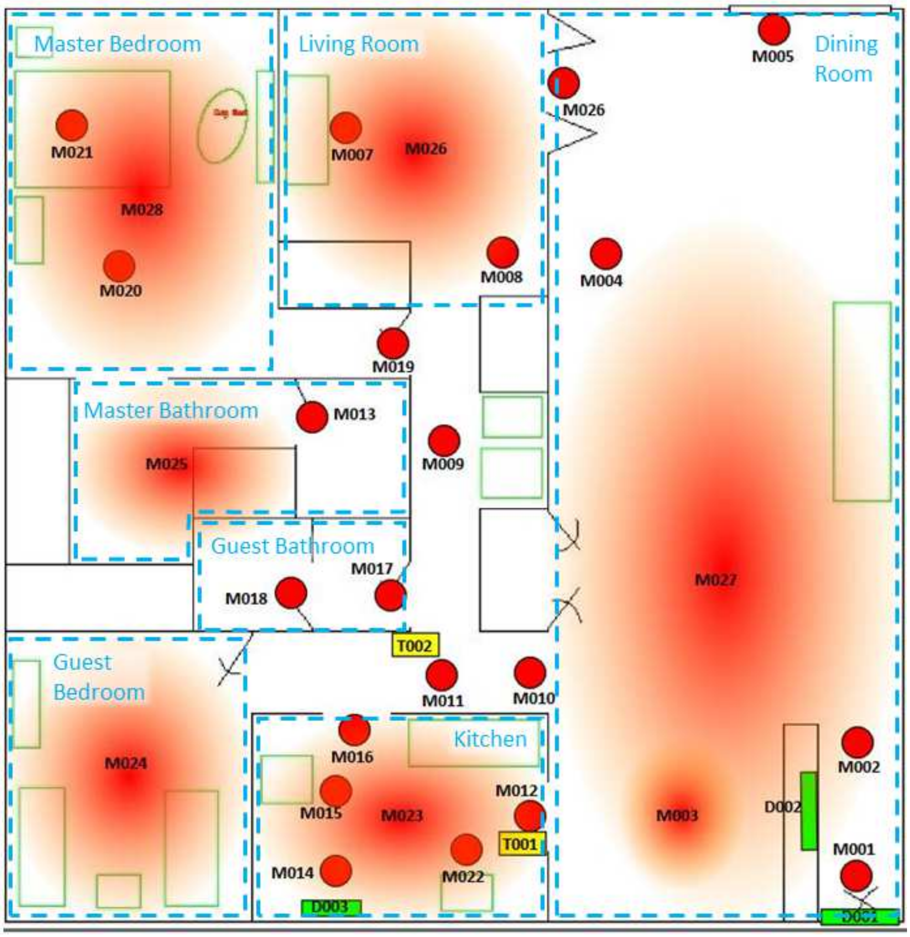

Figure 3.1: The map of the Milan dataset house with marked the installed sensors in the various rooms.

Sensors have collected data about the daily activities performed by the user, in the form of an event log, where each event represents a measurement by a sensor and is described by the record $\langle timestamp, sensor, value \rangle$:

- *timestamp*, which identifies when an event is recorded. It is the date and time when the sensor is triggered due to a user’s action;
- *sensor*: it is the unique identifier of the sensor. We can see on the picture the sensors labeled with a letter, representing the type of the sensor, followed by a numeric sequence.
- *value*, which is the value measured by the sensor; For example it can be

an ON/OFF state, a temperature value or an OPEN/CLOSE state of a door.

Noteworthy that, beside the pet, sometimes guests come to visit the main user. Since the guest and the pet interact with the surrounding environment, triggering home sensors, they represent a real source of noise of the case study.

During the project, the user has been asked to annotate the activity she has been performing, e.g. reading, watching TV, taking medicines and so on. Each of these activity, from now on called *macro activity*, could be represented as a sequence of low level activities that can be measured by sensors (i.e., *micro activities*). For instance, when cooking, the user moves towards the kitchen light switch to turn it on, moves to fridge and after a while also switches on the oven to warm the food, and so on until the user leaves the kitchen. The user must manually specify when she starts and ends the performed macro activity, i.e., the macro activity cooking starts with the user entering the kitchen and ends when she leaves the room. Each of activities in the previous example triggers a big amount of sensors, namely micro activities. Noteworthy that some macro activities could be performed in parallel, e.g., taking medicines while watching TV. In this case, we have that a macro activity begins before the end of the previous one. From a preliminary analysis of data, it seems that macro activities are stored in strict sequence; except rare cases that represent parallel activities or are due to annotation errors. We will discuss this point in Section 3.4.1.

The macro activities performed by the user and that are in the dataset are the following: `Bed To Toilet`, `Chores`, `Desk Activity`, `Sleep`, `Read`, `Dining Room Activity`, `Morning Meds`, `Evening Meds`, `Meditate`, `Guest Bathroom`, `Kitchen Activity`, `Leave Home`, `Master Bathroom`, `Watch TV` and `Master Bedroom Activity`. Figure 3.2 describes an excerpt of the dataset. The dataset is characterized by non-uniform daily distribution of macro activities, as is shown in Figure 3.3. In fact, the average number of macro activities performed by the user in a day is around 41 with a standard deviation of 13.7.

3.3 Preliminaries

We introduce here some basic notation that will be used in the following. Let us denote by A the set of activities and by R the set of resources (e.g., actors). In this work we refer to a quite general notion of process, defined as follows.

Definition 1 *(Process, process instance) A process consists of a flow of activities that are performed by some resource to reach a certain goal. We define a process instance every single execution of a process.*

timestamp	sensor	value	macro activity
...	...	...	...
2009-10-16 08:45:38.000076	M016	ON	Kitchen_Activity begin
2009-10-16 08:45:39.000094	M003	ON	NULL
2009-10-16 08:45:40.000041	M015	ON	NULL
2009-10-16 08:45:40.000090	M023	ON	NULL
2009-10-16 08:45:41.000000	M011	OFF	NULL
...	...	...	...
2009-10-16 08:58:52.000004	M016	ON	Kitchen_Activity end
2009-10-16 08:58:52.000053	M023	OFF	NULL
2009-10-16 08:58:54.000040	M011	ON	NULL
...	...	...	...
2009-10-16 08:59:02.000054	M019	ON	Chores begin
2009-10-16 08:59:04.000072	M019	OFF	NULL
...	...	...	...
2009-10-16 09:14:47.000090	M006	ON	Chores end
...	...	...	...

Figure 3.2: An excerpt of the input dataset. The record $\langle timestamp, sensor, value \rangle$ is automatically gathered by sensors; the feature *macro activity* is manually added.

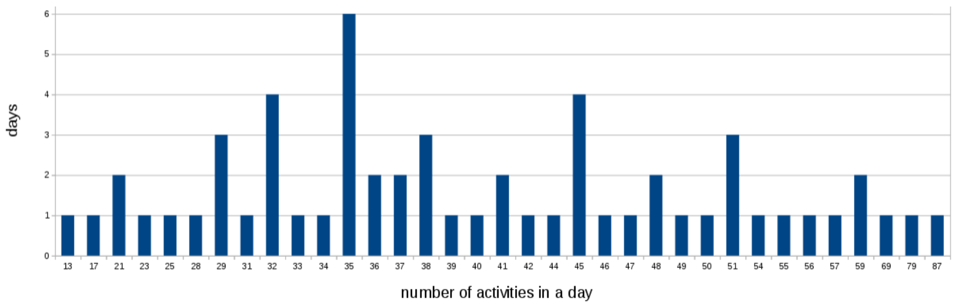

Figure 3.3: Daily distribution of macro activities.

In order to formally characterize a process instance, we provide in the following the notion of event, trace and event log. We refer to a simple notion of event, which can be however extended with further properties.

Definition 2 *(Event) An event σ is a tuple $\langle a, r, t \rangle$, where $a \in A$ is an activity performed by a resource $r \in R$ at a certain timestamp t.*

For instance, the tuple ⟨ Kitchen_Activity begin, Ada,'2009-10-06 08:45:38.000076'⟩, with *Kitchen_Activity begin*$\in A$ and *Ada* $\in R$, represents a valid event.

An *event log* is the dataset recording all past executions (i.e., instances) of a process. Each instance can include parallel execution of activities. However, event logs store only the *trace* of a process instance, i.e., the sequence of the activities stored according to their temporal order of occurrence.

Definition 3 *(Trace, event log) A trace for a certain process instance is a finite sequence of events $l = \{\sigma_1 \ldots, \sigma_n\}$, where $\forall \sigma_i, \sigma_{i+1}, t_i < t_{i+1}$. An event log L*

	activity	resource	timestamp
σ_1	a	r_1	2017-11-10 10:24:43
σ_2	b	r_1	2017-11-10 10:57:19
σ_3	c	r_1	2017-11-10 11:09:32
σ_4	e	r_1	2017-11-10 12:23:54
σ_5	a	r_2	2017-11-10 8:15:23
σ_6	d	r_2	2017-11-10 8:26:08
σ_7	e	r_2	2017-11-10 8:54:34
σ_8	a	r_1	2017-11-11 9:30:15
σ_9	c	r_1	2017-11-11 10:02:08
σ_{10}	b	r_1	2017-11-11 10:43:27
σ_{11}	e	r_1	2017-11-11 11:31:35

is a collection of traces, where each event in a trace is unique, i.e. $\forall \sigma_i, \sigma_j \in L, i \neq j \rightarrow \sigma_i \neq \sigma_j$*.*

To make an example, let us suppose a process related to the morning activities performed in a house, with a set of activities $A = \{a, b, c, d, e\}$ and a set of resources (i.e., people living in the house) $R = \{r_1, r_2\}$. Let us consider the following traces which include the sequence of events as recorded by the sensors: $l_1 = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, $l_2 = \{\sigma_5, \sigma_6, \sigma_7\}$ and $l_3 = \{\sigma_8, \sigma_9, \sigma_{10}, \sigma_{11}\}$ where each event σ_i is defined as in Table 3.3.

From such an event log it is possible to recognize that every instance of process (as recorded by a corresponding trace), starts with activity a and ends with activity e. The sequence of activities between those two points seems to differ from trace to trace, although some regularities can be spotted e.g., c and b always appear together. In general, real-world event logs include thousands of single events and a manual analysis is not feasible. For such a reason, a much better way to look through the possible process flows is a process model.

Processes are frequently represented through *models* representing the control-flow of the performed activities, that is their ordering relation. Among the several formalisms to represent process models, in this paper we refer to Petri nets.

Definition 4 *(Petri net) A Petri net is a tuple* $\langle P, T, F\rangle$*, where* P *is the set of places,* T *the set of transitions such that* $P \cap U = \emptyset$*, and* $F : (P \times T) \cup (T \times P)$ *the flow relation. A firing sequence for a Petri net* N *is a sequence of transitions.*

If a process model is not available for the process at hand, Process Mining techniques can be applied to derive a model from the event log. In Figure 3.4, we show the schema for the morning activity process, represented through a Petri net. Please note that circles represent places, while transition are represented as boxes. In particular, white boxes corresponds to events in the

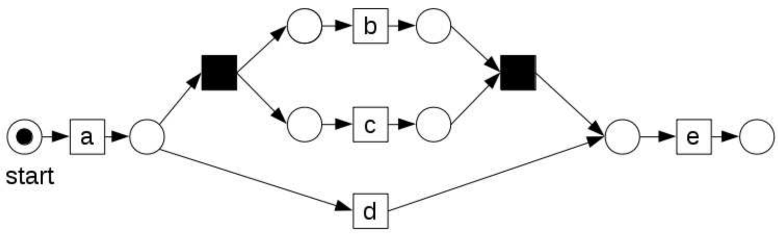

Figure 3.4: Petri net for the example.

log, while black boxes are invisible transitions, which do not correspond to any specific event but are needed to obtain a sound net.

Among the most common parameters used to evaluate the quality of the discovered process model, a relevant one is the *fitness*. It evaluates the ability of the model to replay traces within the log. A fitness value close to 1 means that each trace in the log can be associated with an execution path specified by the process model, while lower values are typical of models that are not able to represent some traces or pieces of them. This parameter is a key aspect in the model quality evaluation since an unfitting model is not able to represent all the recorded events. In order to compute the fitness, conformance checking algorithms [103] are employed to compare the log with the resulting discovered model.

Definition 5 *(Conformance checking) Given a Petri net N and an event log L, conformance checking techniques verifies whether N fits L. N perfectly fits the net N iif each trace $l = \{\sigma_1, \ldots, \sigma_n\} \in L$ can be replayed on the model, i.e. l is a firing sequence for N.*

3.4 Methodology

The previous introduced definition allow us to consider daily macro activities performed by a person at home as the execution of a process, whose model describes the behavior of the user at home (or *daily behavior model*). The sequence of sensors triggered during the execution of a macro activity can be viewed as the execution of a process as well: the related process model represents the model of the macro activity in terms of micro activities, namely *macro activity model*. For these reasons, in this work we exploit Process Mining techniques to obtain both the daily behavior model and macro activities models. To this end, in the next subsection we discuss preprocessing operations, which are needed to handle possible user's annotation errors and to properly set up the data for applying specific Process Mining techniques, which are explained in subsection 3.4.2.

3.4.1 Pre-Processing

As first step, we fixed issues related to manual annotation of macro activities. In particular, we noted that in the dataset there are some macro activities that have started before the end of another (i.e., the timestamp of the begin of activity B is before the timestamp of the end of activity A). This situation can occur in two cases:

parallel activities the two macro activities have been performed in parallel by the user. An example is given in Figure 3.5, where the user watches the TV while is cooking;

sequential activities A and B can not be executed in parallel by their nature; then the user made an annotation error. For instance, the user can not do any other macro activities while is sleeping, so `Bed to Toiled` and `Sleep` must for their nature be performed in sequence (see Figure 3.6).

In the second case, we added artificial events in the log in order to fix the issue. The artificial end of A is added just before the reported begin of B and the artificial begin of A is added just after the reported end of B. Figure 3.7 shows how the wrong parallel behavior of Figure 3.6 has be transformed in a correct sequential behavior.

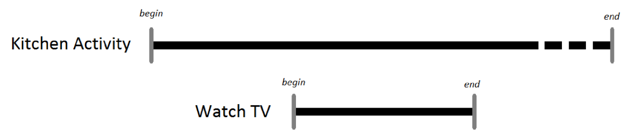

Figure 3.5: An example of real parallel activities.

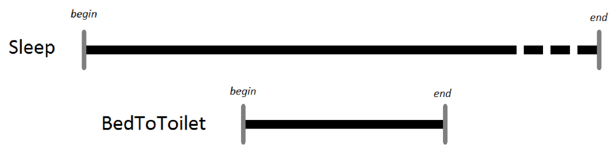

Figure 3.6: An example of wrong parallel activities.

Pre-processing phase produces two datasets suitable for the extraction of the two models: *macro activity event log* and *micro activity event log*. The former

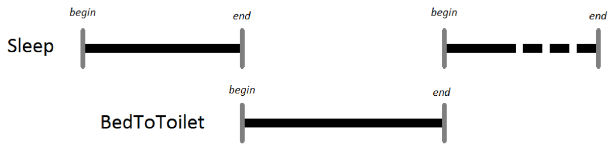

Figure 3.7: An example of wrong parallel activities that have been fixed adding artificial events.

Day 1:	[Sleep, 2009-10-16 03:58:44.068]
	[Morning_Meds, 2009-10-16 08:42:01.077]
	[Watch_TV, 2009-10-17 08:43:59.024]
	[Kitchen_Activity, 2009-10-16 08:45:38.000]
	...
Day 2:	[Sleep, 2009-10-16 21:03:28.000]
	[Bed_to_Toilet, 2009-10-16 21:29:11.000]
	[Bed_to_Toilet, 2009-10-16 22:20:15.000]
	[Sleep, 2009-10-16 22:21:23.000]
	[Morning_Meds, 2009-10-17 09:17:51.000]
	...

Table 3.1: An excerpt of macro activity event log.

is used as input to the process mining phase to obtain the daily behavior model and latter to extract the macro activity model for each macro activity.

In order to obtain the macro activity event log, the first step is to remove from the dataset unnecessary data, namely sensor id and value. Next is aimed at identifying the begin and end of each trace in the event log. Since we are interested in daily activities, the issue becomes to identify the timestamp when a day starts/ends. Using a fixed time to identify a day could lead to split a macro activity (e.g. `Sleep`) into two days. For this reason, we have chosen to set the end of a day dynamically, when the user went to sleep. However, sometimes she went to sleep after 12pm, hence in a new day. Furthermore, sometimes she wakes up during the night to go to the toilet, even several times per night; hence, we have several sleeping activities in a day. For these reasons, we defined the trace related to a given day *D* as the period between the time when the user goes to sleep for the first time after 9pm of the day *D-1* and the first sleeping activity after 9pm of the day *D*.

An excerpt of the *macro activity event log* obtained at the end of the pre-processing is reported in Table 3.1. A day is composed by many rows, each representing the begin of a given activity. Of course, more than once instance of the same activity could be registered during the same day, because the user can perform the same activity many times.

As concerns micro activity, the input dataset has been divided into several parts; producing a micro activity event log for each macro activity. The structure of each micro activity event log is similar to the one reported in Figure 3.2, where the *macro activity* field has been deleted and an identifier representing the different instances (i.e. traces) of the macro activity has been added. With the aim to minimize noise only motion sensors have been considered, in fact temperature and door sensors have been excluded from the log.

Each motion sensor can record an ON or OFF event. The former means that a movement is detected within the sensor's range, while the latter means that after an ON event no further movements have been detected. This behaviour can produce noise when either the user stops within the detection area or she moves very fast. In the first case, the sensor stores an OFF value in the event log, but the user is still in the detection area. As the user starts moving again, the sensor will detect another ON event followed by an OFF event, and so on until the user leaves the area (see Figure 3.8). This produces in the event log a sequence of ON and OFF values of the same sensor. In the second case, if the sampling rate is set quite slow, the user could leave the detection area before the OFF event is stored. Hence, in the event log there could be the activation of a new sensor before the OFF of the previous one. In order to reduce noise, we removed the OFF events from each micro activity event log. Furthermore, information conveyed by OFF events is related to the time when the user ends a micro activity, hence it is not relevant for the goal to detect the process model of activity performed by the user.

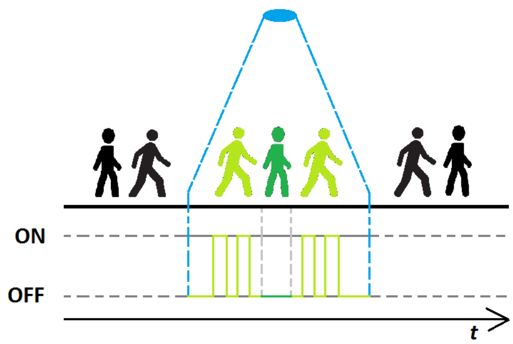

Figure 3.8: IR motion sensors.

In order to have unique start and end points for any instance of the process, for each trace of all event logs, artificial events corresponding to "START" and "END" activities have been added as first and last event respectively. This widespread practice is aimed at improving the soundness of Petri net resulting by a Process Discovery algorithm [98]. This operation is particularly important when analysing micro activity event logs. Indeed, it is noteworthy that different traces in the same micro activity event log could have different activities in the first and last position. This is due to: (a) different ways (i.e., process variants) a macro activity is executed (e.g., `Kitchen Activity` could start triggering the sensor closed to one of the two kitchen doors, i.e., M012 and M016), and (2) the time shift between the sensor activation and the manual annotation of macro activities.

3.4.2 Process Discovery

Amongst the various process mining techniques, in this section we will discuss about process discovery algorithms which are aimed at the extraction of process models starting from sensors' event logs. In particular, among of the most important discovery techniques there are: Alpha Miner [103], Fuzzy Miner [104], Heuristic Miner [105], and infrequent Inductive Miner [98].

The Alpha Miner algorithm has been designed for discovering process models starting from "clean" conditions, that is noiseless logs. It gives good results with structured processes (known in Literature as *lasagna-like* processes), whereas it fails in the discovery valid models for highly variable processes (known as *spaghetti-like* processes) returning overgeneralizing "flower" models, which are not useful knowledge as any behavior is represented (see [106]).

The dataset we are dealing with is characterized by a high standard deviation value, as we can see from Figure 3.3. This fact suggests that we are dealing with a log representing a highly variable process. Because of the previous motivation, we might think to adopt the Fuzzy, Heuristic or infrequent Inductive Miner algorithms.

The Fuzzy Miner algorithm filters and groups the most infrequent behaviors providing as output a high level (i.e., abstract) representation of the extracted model. The outcome of such an algorithm is a scheme involving just the most relevant activities, displayed as single activities or aggregated in clusters, and their precedence relations. Moreover we should notice that Fuzzy Miner is not capable to represent split and join constructs. It means that the resulting models do not have an executable semantics. The Heuristic Miner can be thought as an extension of the Alpha Miner algorithm. The last one is able to find all causal relations amongst activities within the log, whereas the Heuristic Miner takes into account the frequency of relations and applies some heuristics to determine relevant sequence and parallelism relations. Finally, infrequent Inductive Miner returns a model by iteratively refining a process tree, where each node represents either an activity of the process (leaf node), or an operator (branch node). As for Heuristic, also this algorithm filters infrequent behaviors and the output can be represented by a Petri net. However, we like to note that infrequent Inductive Miner guarantees to always return a sound model, i.e. a model where all process steps can be executed and the final marking of the Petri net is always reachable. Depending on the previous considerations, we have decided for the adoption the infrequent Inductive Miner Algorithm [98]. The filtering level of the infrequent Inductive Miner depends on a user-defined threshold, which ranges from 0 to 1; the value of 0 indicates that the infrequent paths are not filtered at all. Depending on the previous considerations, we have decided for the adoption of the infrequent Inductive Miner algorithm [98]. In particular the experiments (see Sections 3.4.3 and 3.4.4) have been performed using the algorithm's implementation provided within ProM Tools[2], a well-known framework which implements several process mining algorithms through a modular plugin system.

Next subsections present experiments performed to extract the daily behavior model and the macro activity models respectively.

Amongst the most common parameters used to evaluate the quality of the discovered model, a relevant one is the *fitness*. It evaluates the ability of the model to replay traces within the log. A fitness value closed to 1 means that each trace in the log can be associated with an execution path specified by the process model, lower values represent a model that is not able to represent some traces or pieces of them. This parameter represent a key aspect in the model quality evaluation since an unfitting model is not able to represent all the recorded events. In order to compute the fitness, conformance checking algorithms [103] are employed to compare the log with the resulting discovered model.

3.4.3 Daily Behavior Model

In this Section we present results of experiments aimed at discovering the daily behavior model of the user living at home. To this end, the infrequent Inductive Miner [98] algorithm has been used to elaborate upon the *macro activity event log*.

In Figure 3.9 we can see the fitness trend obtained by setting different values of the threshold parameter. Low threshold values correspond to a model characterized by the fact that almost all causal relations are considered.

[2] http://www.promtools.org

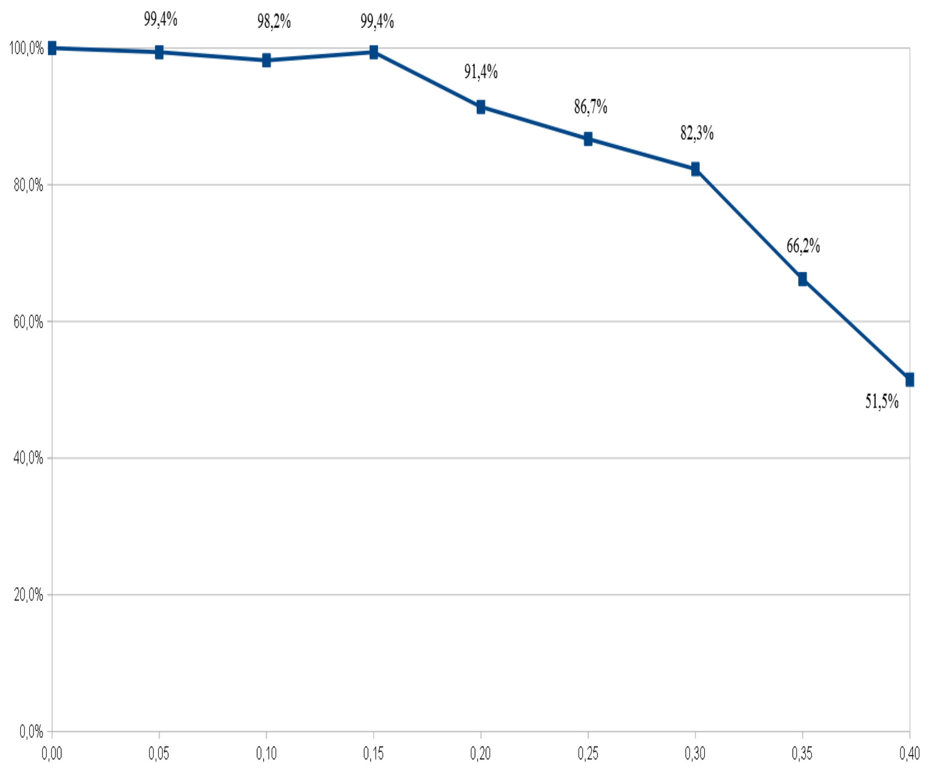

Figure 3.9: The fitness obtained by using the infrequent Inductive Miner with different values of the threshold parameter.

In such a case, we obtain a model with high fitness value but characterized by a big amount of transitions, places and arcs, hence a model that might be complex to interpret. On the other hand, high threshold values lead to a very simple, quite trivial, model which does not bring any contribution to the knowledge. During the experiments, it has been found that a good compromise can be obtained setting up the threshold value to 0.20. By using this setting, in fact, the fitness of the discovered model is 91.4% and the model is quite simple (see Figure 3.10[3]).

[3]A high-resolution version is available at http://kdmg.dii.univpm.it/?q=content/daily-behavior-model

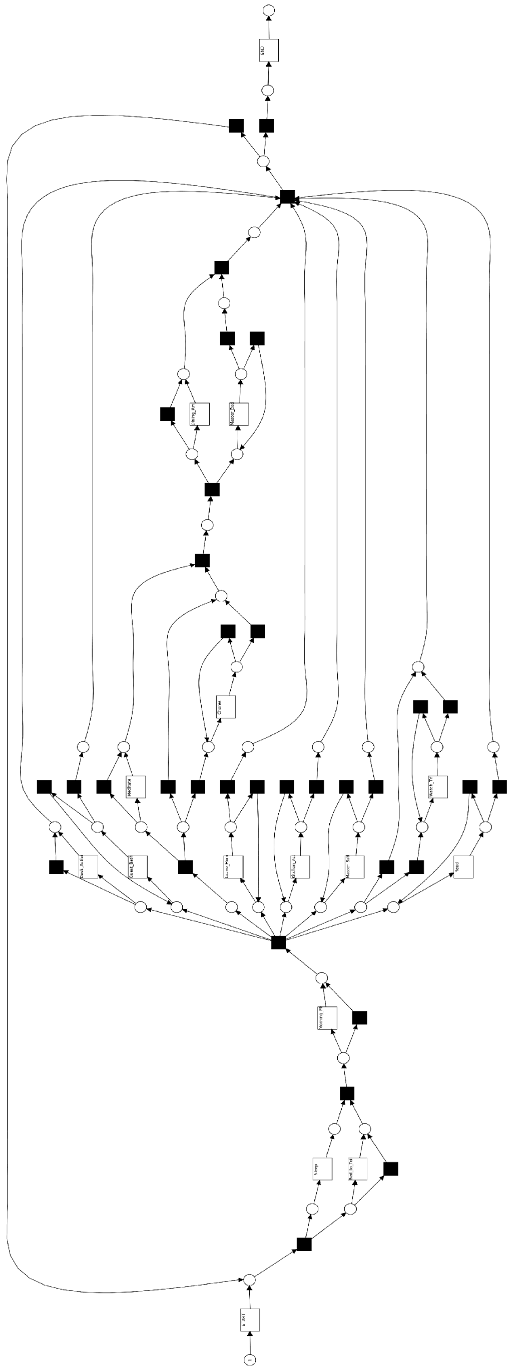

Figure 3.10: Petri net of the Daily Behavior process model obtained with a threshold value of 0.20 of the Infrequent Inductive Miner.

In order to give an example, Figure 3.11 focuses on the first part of the model, which corresponds to the way in which the monitored user begins her typical day. From that excerpt, we can notice that sometimes during the night the user stops her sleeping for going to the toilet, and then comes back in the bedroom for sleeping again. We see also that after waking up the user usually takes her morning medicines. Then several scenarios are possible, which allow us to derive useful insights. For instance, `Kitchen Activity` and `Master Bathroom` usually are not in sequence, while `Master Bedroom Activity` and `Dining Room Activity` usually occur after `Meditate` and/or `Chores`.

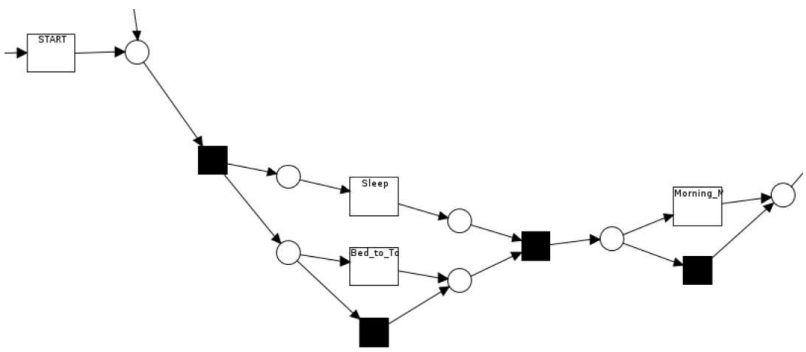

Figure 3.11: An excerpt of the Daily Behavior process model.

3.4.4 Macro Activity Model

This Section is devoted to present results of experiments aimed at discovering macro activity models. As we have explained we remember that macro activities represent the flow of micro activities (i.e., sensors) triggered when a macro activity is performed. Without loss of generality, we focus on some of relevant macro activities. For each of these macro activities, Table 3.2 show some statics of related micro activity event logs, namely the number of traces per log and the average number of events per trace.

From the Table, it turns out that each trace is characterized by several events, namely triggered sensors. In order to reduce the complexity of the problem and the noise caused by infrequent sequence of sensors' activations (mainly due to the pet and guests that sometimes visit the main user), for each macro activity we only consider most frequently triggered motion sensors. In particular, for each micro activity event log, we keep only those sensors whose occurrence frequency is greater or equal than 2% of the corresponding event log. Table 3.4 shows these sensors and related frequencies. After this filtering operations, we obtain a set of *reduced logs* for the considered macro activities. These logs have been used to extract macro activity models adopting the infrequent Inductive

activity	**number of traces**	**avg events per trace**
Watch TV	114	207.79
Kitchen Activity	554	232.75
Read	314	160.13
Desk Activity	54	141.26
Meditate	17	77.35
Master Bathroom	306	49.25
Master Bedroom Activity	117	233.65
Guest Bathroom	330	32.12

Table 3.2: Number of traces and average number of events per trace in micro activity event logs used in the experiments.

Miner algorithm, where the threshold parameter is set to 0.20. Discovered models show a high fitness with respect to related reduced logs, as reported in Table 3.3. This means that each model is able to correctly replay its reduced log.

In order to evaluate the effectiveness of models in describing the related macro activity, we have also computed the fitness of each macro activity model with respect to any micro activity event logs. Results are shown in Table 3.5.

activity	fitness (%)
Watch TV	99.69
Kitchen Activity	97.96
Read	94.15
Desk Activity	94.91
Meditate	99.69
Master Bathroom	99.49
Master Bedroom	98.56
Guest Bathroom	93.82

Table 3.3: Fitness values obtained by checking the conformance of macro activity models over related reduced logs.

The model discovered for a certain macro activity a is an effective representation if it is able to replay the traces in the corresponding micro activity event log and is not able to replay traces of other event logs. This means that, for the i-th row of Table 3.5 corresponding to a certain macro activity a_i, the maximum fitness value should be at the i-th column of the event log (in bold in the Table), while values for other columns should be lower. This behavior is experimentally confirmed for the following macro activity models: `Kitchen Activity`, `Read`, `Desk Activity`, `Meditate`, `Master Bathroom` and `Guest Bathroom`. Models obtained for `Watch TV` and `Master Bedroom Activity` return high fitness values both for the related activity and for another one. In particular, the `Watch`

Watch TV		**Kitchen Activity**		**Read**		**Desk Activity**	
Sensor	Freq.	Sensor	Freq.	Sensor	Freq.	Sensor	Freq.
M008	48.08%	M023	26.86%	M004	65.67%	M007	63.38%
M026	8.16%	M014	17.26%	M027	5.61%	M026	15.78%
M003	2.98%	M022	14.24%	M003	3.79%	M008	2.83%
M007	2.96%	M015	11.05%	M001	2.28%		
M001	2.87%	M012	7.24%	M012	2.25%		
M027	2.53%	M027	3.84%	M022	2.15%		
M006	2.10%	M003	3.67%	M005	2.05%		
M017	2.03%						
Master Bedroom Act.		**Meditate**		**Master Bathroom**		**Guest Bathroom**	
Sensor	Freq.	Sensor	Freq.	Sensor	Freq.	Sensor	Freq.
M028	25.35%	M024	62.66%	M025	51.46%	M018	38.76%
M025	23.83%	M011	7.00%	M013	23.04%	M017	18.37%
M020	12.06%	M028	3.65%	M028	8.85%	M003	6.17%
M021	7.08%	M016	3.19%	M020	3.86%	M010	6.08%
M013	4.53%	M025	2.89%			M011	5.74%
M019	4.04%	M020	2.13%			M009	3.43%
M009	2.81%	M022	2.05%			M016	2.45%
						M022	2.42%
						M012	2.24%
						M015	2.16%

Table 3.4: Sensors and related occurrence frequencies. For each micro activity event log, only sensors with frequencies greater or equal than 2% are reported.

Reduced Models	Micro Activity Event logs			
	Watch TV	**Kitchen Activity**	**Read**	**Desk Activity**
Watch TV	**77.30**	14.73	17.15	89.61
Kitchen Activity	14.29	**88.42**	19.37	8.20
Read	7.30	19.57	**80.55**	7.98
Desk Activity	13.49	6.58	6.02	**82.70**
Meditate	8.97	23.14	7.03	8.27
Master Bathroom	5.77	6.81	5.66	8.00
Master Bedroom	8.77	7.37	5.88	9.48
Guest Bathroom	11.79	20.00	7.75	7.37
	Meditate	**Master Bathroom**	**Master Bedroom**	**Guest Bathroom**
Watch TV	11.60	15.17	13.24	34.77
Kitchen Activity	17.09	13.72	10.01	24.96
Read	10.71	14.65	7.63	18.67
Desk Activity	8.35	14.31	6.74	15.85
Meditate	**83.80**	53.54	56.29	23.33
Master Bathroom	11.30	**90.72**	67.40	15.52
Master Bedroom	12.64	86.85	**82.32**	16.89
Guest Bathroom	14.15	15.56	11.46	**81.15**

Table 3.5: Fitness values obtained by replaying all micro activity event logs over all macro activity models.

`TV` model seems to be able to better represent the event log of `Desk Activity` than the one of `Watch TV`. The main reason for this inaccurate result is that the two activities take place in the same room and in both cases the user is often stopped. As a matter of fact, the two event logs share the most frequent sensors, i.e., M007, M008 and M026. Furthermore, the reduced log for `Watch TV` includes all sensors in the reduced log of `Desk Activity`. Hence, the model for `Watch TV` is able to replay traces of `Desk Activity`, but the viceversa does not hold. We observe the same behavior for `Master Bedroom Activity` and `Master Bathroom`. In this case, the reduced log of `Master Bedroom Activity` contains sensors of the one of `Master Bathroom`. When we check the conformance of the micro activity event logs of `Master Bathroom` and `Master Bedroom Activity` over the `Master Bedroom Activity` model, the resulting fitness values are 86.85% and 82.32% respectively. Replaying the same logs over `Master Bathroom` model, fitness values become 90.72% and 67.40%, as expected. In order to improve the results one could reduce the granularity (e.g., by merging `Desk Activity` and `Watch TV` in a sort of living room activity) or the filtering frequency for those micro activity event logs with few sensors in the reduced logs (e.g., `Desk Activity` and `Master Bathroom`).

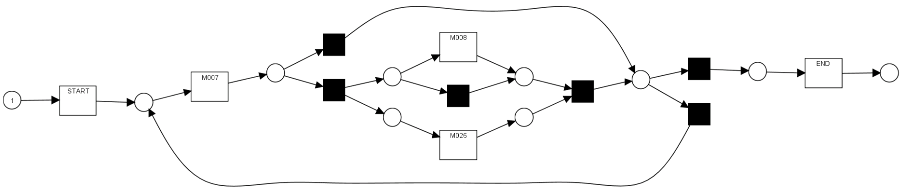

Figure 3.12: Macro activity for `Desk`.

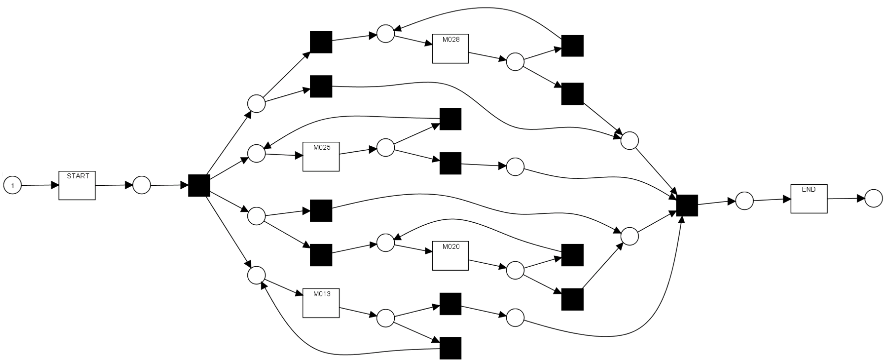

Figure 3.13: Macro activity for `Master Bathroom`.

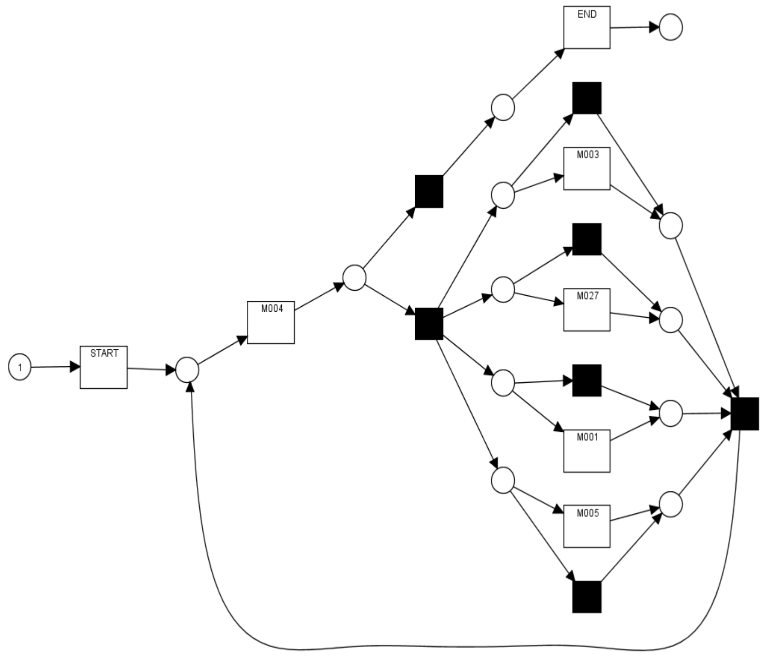

Figure 3.14: Macro activity for `Read`.

For the sake of space, in Figures 3.12, 3.13 and 3.14 we show some of the eight discovered models[4], namely `Desk Activity`, `Master Bathroom` and `Read` respectively.

The three models highlight how the use of the reduced log allow to simplify the discovered model and to make them more understandable to machines as well as human analysts. The model for the `Desk` macro activity (Figure 3.12) is formed only by living room sensors (see Figure 3.1). In particular, the process always starts when the sensor M007 is triggered, which is the sensor closed to the desk. Then the process involves either the cyclic activation of M007 (the upper and lower paths of the model) or the activation of the other two sensors of the room (middle path of the model). The former case is due to the peculiarity of the activity; as a matter of fact, the user spends most of the time sitting in the chair, but she is not perfectly stationary and several subsequent activations of the sensor are stored in the event log (see Section 3.4.1). In the latter case, as one can expect, if M008 is triggered then also M026 is activated; in fact, to activate M008 the user has to mote through the middle of the room where M026 is. As concerns the `Master Bathroom` macro activity model of Figure 3.13, the activation cycles of sensors (M018, M020, M025 and M028) are more

[4] All models are available at http://kdmg.dii.univpm.it/?q=content/macro-activity-models.

evident. The reason for this behavior is due to the fact that most of time the user moves while she is performing bathroom tasks, e.g., shower, wash and so on. It is noteworthy that M013 and M025 are activated in parallel; this means that every time the user enters the room the two sensors are triggered without a prevailing order among them. The other two sensors (M020 and M028) are triggered only in some execution paths. This happens since the user going to the bathroom passes by the bedroom, hence sometimes is within the detection areas of M020 and M028. The use of reduced log in simplifying the discovered model is even clearer in the macro activity model of `Read` (Figure 3.14). In the main path (the upper in the model) only the M004 sensor is activated, showing that the user stands still most of the time as she reads. Sometimes she moves while reading, so other sensors in the room are triggered.

3.5 Related Work

Most work in the Literature dealing with user behavior analysis in smart environments refer to domotic scenarios in the context of smart homes. In the Ambient Assisted Living (AAL) field, some work in the last years has focused on recognizing user activities of daily living (ADL) [107], starting from raw data collected through a set of sensors. AAL community classifies this work into two different groups: Activities of Daily Living and Instrumental Activities of Daily Living (IADLs) [108]. The main difference is that in IADLs there is an interaction between the user and electronic devices (e.g. phone, home appliances), contrary to ADLs where there is not such kind of an interaction, like in the proposed approach. In [109] authors propose an approach aimed to predict a macro activity from values collected by simple and cheap sensors. The prediction is performed by using a Naive Bayes classifier. Naive Bayesian classifiers assume that class attributes are not dependent given the class. Furthermore, it is assumed that all attributes impacting on a classification decision are observable. Since these reasons, Naive Bayesian classifiers are known to not give good results in real-world contexts, although experimental results show they work well on some domains. In fact, results demonstrates that ubiquitous, simple sensor devices can be used to recognize activities of daily living. The resulting classification models are not able to describe the user behavior, but only recognize that a trace is an execution of a given activity.

In [110], authors propose an unsupervised approach aimed at discovering interesting patterns of ADLs. The introduced solution is able to identify a pattern independently of the time intervals between the activations of two sensors. We highlight that the approach we propose in this work is time independent too, because timestamps are only used to identify an ordering among activities. A further contribution from D.J. Cook and her research group in the field of activity recognition area, is described in [111]. In their article authors explain how monitoring ADL at home can be exploited to predict clinical score of inhabitants.

The approach proposed in [112] is able to identify patterns and segment data streams into patterns that correspond to activities, considering also unexpected changes in the users' behavior and abnormalities occurred within a specific threshold, as typically happens with Alzheimer's patients. A frequent pattern mining algorithm is here applied in order to find patterns within the stream of captured events.

Another methodology is introduced in [113], where authors propose to recognize daily routines as a probabilistic combination of activity patterns. They show how probabilistic topic models can be exploited for the automatic discovery of patterns.

In the last years, researchers have started to address the problem of the activity recognition with a novel approach which is based on exploiting Process Mining techniques and algorithms [114, 115, 116, 117]. These approaches basically share the fact that every activity executed by a user can be seen as a step of a process instance. In particular, in [114], the use of Process Mining to fulfill several tasks is introduced. Such an approach shares with this work the analysis of user behavior, defined as a process model of (macro) activities, where activities are characterized by users through smartphones and smart watches. The adoption of personal and/or wearable devices, in particular, provides more precise data to track user behavior, minimizing the presence of noise due to the activation of multiple sensors or the presence of more than one user.

Zakwan et al. [118], address the problem introducing a formal approach based on an ontology that is devoted to represent user activities. It also describes a process model representing standard/expected user behavior. Process Mining and conformance checking techniques are then used for model discovery starting from sensor data and for comparison purposes.

In [115], authors proposed a system to analyse the behavior of health staff in a surgical area of an hospital. As initial step, an indoor location systems (also known as realtime location systems (RTLS)) is used to collect data, that are then assigned to a (macro) activity. In the end, Process Mining techniques are exploited to discover a model able to represent and describe the staff behavior. The authors developed a specific tool designed to use Process Mining techniques with indoor location system data. In particular, they adopted the PALIA [119] algorithm which exploits syntactical pattern recognition techniques to build a formal automaton, known as Timed Parallel Automaton (TPA) [120], i.e. a safe Petri net that can represent the process.

Senderovich et al. [116] propose an interesting approach focused on a preprocessing stage aimed at mapping sensor data to event logs based on domain knowledge. In particular their approach is devoted to transform precision raw sensor measurements containing locations and timestamps of process entities into standardized event logs comprising process instances. To this aim, the authors introduce the definition of "interaction" which acts as intermediate knowledge layer.

Another contribution in the field of activity recognition comes from the research of Mecella et al. [117]. They address the problem of the identification of habits related to multiple inhabitants living within the same house, proposing a solution based on the Fuzzy Miner [104] algorithm. In contrast to presented methodology, all the above mentioned work do not deal with the discovery of macro activity models that can be extracted from sensor data.

Some work in the Literature focuses on more generic smart environments, such as smart offices of smart factories. For instance, in [121] a Knowledge Discovery approach is adopted for extracting behavioral patterns for occupants in offices, mainly with the aim to characterize energy performances of buildings. Here, collected data refer to window opening and other activities, and applied techniques include clustering, decision tree models and rule induction algorithms.

Chapter 4

Further Ph.D. Activities

In this Chapter some side projects and activities carried on during the Ph.D are briefly described. In particular in the field of Educational Data Mining [122]. Data mining techniques can be adopted on a big number of digital resources with the aim of extracting knowledge of the most various facets related to learning processes. When bachelor and master degrees were introduced in the Italian system, the Engineering Faculty of Università Politecnica delle Marche removed mandatory prerequisites from its degrees. If course A is defined as a prerequisite for course B, then A's contents are considered relevant to understand B's contents and then they should be acquired before. Removing mandatory prerequisites means that, although such an information is provided to students, they may take the exam of course B before the exam of course A.

Because of rules modification, intense debates are rised among professors stating such new rule modifications that can lead to unstructured learning practices with consequent performance issues. On the other side, students who are concerned of an overly constrained course of study. Because of this reasons, under the initiative of the Faculty Committee named *Commissione Paritetica* [123] we have decided to perform an ex-post analysis of student's careers in order to analyze the effects of the new rules. In particular, we have proposed a methodology exploiting process mining techniques with the aim to understand the typical paths that the students enrolled in the bachelor degree in Computer and Automation Engineering follow to take exams, and their relations with final outcomes.

After an initial preprocessing stage, we have obtained a set of traces representing students' careers. Traces are classified into three groups according to the final grade mark and length of the period of study. In the next step, exploiting the infrequent Inductive Miner (iIM) algorithm [98], for each group we have extracted a process model in the form of a Petri Net. The accuracy level of the models is assessed by measuring its fitness over the entire set of traces within the group. Results shows a obvious difference between high performance and low performance careers in terms of structure, suggesting that following the recommended prerequisites can favorable.

In Figure 4.1, we can notice in the initial part of the model a well structured flow.

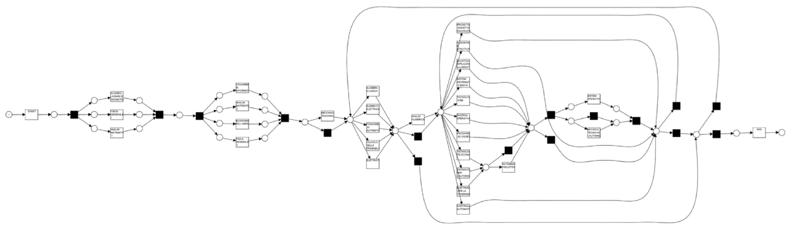

Figure 4.1: Process model of best performing students.

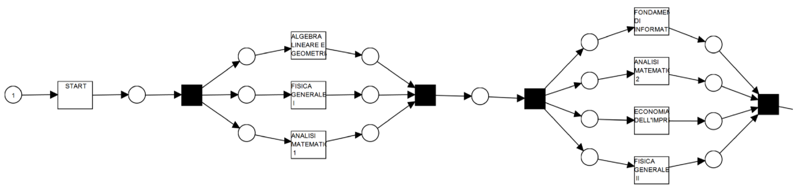

Figure 4.2: Process model of best performing students: detail of the initial part.

As we can notice from a detail in Figure 4.2, the process begins with three parallel events, corresponding to exams taken for courses of Algebra Lineare e Geometria, Fisica 1, Analisi Matematica 1. It means that, although there is not a strict precedence relation among these exams in the log, students usually take all the three before moving to other exams. The three courses are scheduled during the first semester of the first year. In a similar way, the next four parallel events correspond to courses scheduled during the second semester of the first year. Since that, we can notice that students with the best careers basically follow the ideal order in taking exams of the first year.

More detailed and complete results of such a work are available in:

Cameranesi, M., Diamantini, C., Genga L. and Potena D. “Tell the student: evidence-based advantages of prerequisites”. In Proc. of 25th Italian Symposium on Advanced Database Systems. 25-29 June 2017, Squillace Lido, CZ, Italy.

Cameranesi, M., Diamantini, C., Genga L. and Potena D. “Students’ Careers Analysis: a Process Mining Approach”. In Proceeding WIMS ’17 Proc. of the 7th International Conference on Web Intelligence, Mining and Semantics, Amantea, Italy — June 19 - 22, 2017.

Chapter 5

Conclusions

This chapter is devoted to draw some conclusions of the introduced research activity. In particular a discussion are briefly described about the first part of this thesis which focuses on the goal-driven ontology and a discussion about the second part of this work regarding the process mining contribution.

In the first Chapter 2, it has been introduced an ontology describing the AAL domain in terms of goals, tasks, measures and sensors, to support the goal-oriented development process for AAL systems. A reasoning framework based on logic programming aims to retrieve information and to keep the ontology consistent. We have also given a quick overview of possible applications that can be helpful for AAL design processes: this is a part of a larger validation within the HicMO [54] project, with thirteen companies specialized in microelectronic devices, house appliances and telecommunication systems. As a by-product, the ontology provides also a reusable, incrementally built knowledge base for reference.

As future work, beside the applications previously discussed in Chapter 2.5, further usages of GoAAL include context-aware systems for an automated home, in which a sensor network continuously monitors relevant measures that are stored in a repository. In such a context the ontology represents a reference vocabulary which is able to semantically enrich relevant terms of the domain (i.e. sensors, measures and rooms). From an analytical perspective, the presented ontology can also be employed in an ontology-based data access (OBDA) approach, to make queries over the repository using ontology terms and exploiting reasoning functionalities.

In Chapter 3 it has been discussed a methodology aimed at discovering daily behavior model and macro activities models of a user from event logs generated by the activation of sensors in a smart environment. After a preliminary preprocessing step needed to address issues related to errors and environmental noise, Process Mining techniques are exploited for model generation, in particular the infrequent Inductive Miner algorithm [98]. While traditional activity models characterize the way a user executes a sequence of daily tasks (e.g., reading, watching TV, sleeping etc. etc.), the specific focus on micro-activity models is novel in the Literature, and allows to retrieve the flow of sensors activations when a given macro activity is performed by the user.

The efficacy and effectiveness of the approach has been experimentally evaluated by considering a real-world case study from the CASAS project of the Washington State University. Further experimentations are in line, both taking into account other datasets from the same project and from HicMO [54] a project funded by Marche region which involved several enterprises and universities to establish a laboratory for active ageing. Several extensions of the methodology are currently being developed. Firstly, we are studying techniques to detect the most frequent behaviors and we are also planning to exploit subgraph mining techniques (as suggested by [124]). Moreover, the application of this technique to environments with multiple users will require particular care in defining the criteria to distinguish activities performed by each of them separately or collaborative activities performed jointly, which will pave the way to the application of the approach to multi-user scenarios.

Bibliography

[1] Mark Weiser, "The computer for the 21st century," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, July 1999.

[2] Dieter Uckelmann, Mark Harrison, and Florian Michahelles, "An architectural approach towards the future internet of things.," in *Architecting the Internet of Things*, Dieter Uckelmann, Mark Harrison, and Florian Michahelles, Eds., pp. 1–24. Springer, 2011.

[3] K. Asthon, *That "Internet of Things" Thing*, Jun. 2009.

[4] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé, *Vision and Challenges for Realising the Internet of Things*, Publications Office of the European Union, 2010.

[5] Mohammad-Reza Tazari, Francesco Furfari, Álvaro Fides-Valero, Sten Hanke, Oliver Höftberger, Dionisis D. Kehagias, Miran Mosmondor, Reiner Wichert, and Peter Wolf, "The universaal reference model for AAL," in *Handbook of Ambient Assisted Living*, vol. 11 of *Ambient Intelligence and Smart Environments*, pp. 610–625. IOS Press, 2012.

[6] E Aarts and José Encarnação, "True visions. the emergence of ambient intelligence," 01 2008.

[7] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula, "Review: Ambient intelligence: Technologies, applications, and opportunities," *Pervasive Mob. Comput.*, vol. 5, no. 4, pp. 277–298, Aug. 2009.

[8] M. Perry, A. Dowdall, L. Lines, and K. Hone, "Multimodal and ubiquitous computing systems: Supporting independent-living older users," *Trans. Info. Tech. Biomed.*, vol. 8, no. 3, pp. 258–270, Sept. 2004.

[9] Vikramaditya Jakkula, Diane J. Cook, and Aaron S. Crandall, "Temporal pattern discovery for anomaly detection in a smart home," in *The IET International Conference on Intelligent Environments*, 2007, pp. 339–345.

[10] Martha E. Pollack, Laura E. Brown, Dirk Colbry, Colleen E. McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos, "Autominder: An intelligent cognitive orthotic system for

people with memory impairment," *Robotics and Autonomous Systems (RAS)*, vol. 44, no. 3-4, pp. 273–282, 2003.

[11] Yi Chu, Young Chol Song, Henry A. Kautz, and Richard Levinson, "When did you start doing that thing that you do? interactive activity recognition and prompting," in *Artificial Intelligence and Smarter Living*. 2011, vol. WS-11-07 of *AAAI Workshops*, AAAI.

[12] Kevin Curran, Eoghan Furey, Tom Lunney, Jose Santos, Derek Woods, and Aiden McCaughey, "An evaluation of indoor location determination technologies," *J. Locat. Based Serv.*, vol. 5, no. 2, pp. 61–78, June 2011.

[13] Jie Xiong and Kyle Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, Berkeley, CA, USA, 2013, nsdi'13, pp. 71–84, USENIX Association.

[14] Robert J. Orr and Gregory D. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA, 2000, CHI EA '00, pp. 275–276, ACM.

[15] Liang Wang, Tao Gu, Xianping Tao, and Jian Lu, "Sensor-based human activity recognition in a multi-user scenario," in *Proceedings of the European Conference on Ambient Intelligence*, Berlin, Heidelberg, 2009, AmI '09, pp. 78–87, Springer-Verlag.

[16] Yi-Ting Chiang, K. C. Hsu, C. H. Lu, Li-Chen Fu, and Jane Yung-Jen Hsu, "Interaction models for multiple-resident activity recognition in a smart home," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 3753–3758.

[17] Aaron S. Crandall and Diane J. Cook, "Coping with multiple residents in a smart environment," *J. Ambient Intell. Smart Environ.*, vol. 1, no. 4, pp. 323–334, Dec. 2009.

[18] Douglas Perednia and A Allen, "Telemedicine technology and clinical applications," vol. 273, pp. 483–8, 03 1995.

[19] Marco Di Rienzo, Francesco Rizzo, Gianfranco Parati, Gabriella Brambilla, Maurizio Ferratini, and Paolo Castiglioni, "Magic system: a new textile-based wearable device for biological signal monitoring. applicability in daily life and clinical setting," vol. 7, pp. 7167–9, 02 2005.

[20] R. Paradiso, G. Loriga, and N. Taccini, "A wearable health care system based on knitted integrated sensors," *Trans. Info. Tech. Biomed.*, vol. 9, no. 3, pp. 337–344, Sept. 2005.

[21] Stéphanie Pasche, Silvia Angeloni, Réal Ischer, Martha Liley, Jean Luprano, and Guy Voirin, "Wearable biosensors for monitoring wound healing," vol. 57, pp. 80–87, 09 2008.

[22] Winston Wu, Lawrence Au, Brett Jordan, Thanos Stathopoulos, Maxim Batalin, William Kaiser, Alireza Vahdatpour, Majid Sarrafzadeh, Meika Fang, and Joshua Chodosh, "The smartcane system: An assistive device for geriatrics," in *Proceedings of the ICST 3rd International Conference on Body Area Networks*, ICST, Brussels, Belgium, Belgium, 2008, BodyNets '08, pp. 2:1–2:4, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[23] G. Wu and S. Xue, "Portable preimpact fall detector with inertial sensors," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 178–183, April 2008.

[24] C. F. Lai, S. Y. Chang, H. C. Chao, and Y. M. Huang, "Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling," *IEEE Sensors Journal*, vol. 11, no. 3, pp. 763–770, March 2011.

[25] Majd Alwan, P.J. Rajendran, S Kell, David Mack, S Dalal, M Wolfe, and Robin Felder, "A smart and passive floor-vibration based fall detector for elderly," vol. 1, pp. 1003 – 1007, 01 2006.

[26] Xiaodan Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using gaussian mixture models and gmm supervectors," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 69–72.

[27] Fathima Nasreen Amal B. Krishna, Alphy Teresa George, "Home energy management system based on power line communication.," *International Research Journal of Engineering and Technology*, vol. 4, Apr. 2017.

[28] J. J. Baviskar, A. Y. Mulla, A. J. Baviskar, N. B. Panchal, and R. P. Makwana, "Implementation of 802.15.4 for designing of home automation and power monitoring system," in *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on*, March 2014, pp. 1–5.

[29] P. Cottone, S. Gaglio, G. L. Re, and M. Ortolani, "User activity recognition for energy saving in smart homes," in *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*, Oct 2013, pp. 1–9.

[30] P. Rashidi, D. J. Cook, L. B. Holder, and M. Schmitter-Edgecombe, "Discovering activities to recognize and track in a smart environment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527–539, April 2011.

[31] Paolo Bellavista, Antonio Corradi, Mario Fanelli, and Luca Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 24:1–24:45, Sept. 2012.

[32] Mihail Popescu, Benjapon Hotrabhavananda, Michael Moore, and Marjorie Skubic, "Vampir- an automatic fall detection system using a vertical pir sensor array.," in *PervasiveHealth*. 2012, pp. 163–166, IEEE.

[33] L. M. Ni, Yunhao Liu, Yiu Cho Lau, and A. P. Patil, "Landmarc: indoor location sensing using active rfid," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, March 2003, pp. 407–415.

[34] Jeffrey Hightower, Chris Vakili, Gaetano Borriello, and Roy Want, "Design and calibration of the spoton ad-hoc location sensing system," Tech. Rep., 2001.

[35] Kun-Yi Huang, Chi-Chun Hsia, Ming-shih Tsai, Yu-Hsien Chiu, and Gwo-Lang Yan, *Activity Recognition by Detecting Acoustic Events for Eldercare*, pp. 1522–1525, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[36] Lode Vuegen, B Van Den Broeck, P Karsmakers, Jort F Gemmeke, B Vanrumste, and H Van Hamme, "An mfcc-gmm approach for event detection and classification," Tech. Rep., 2013.

[37] Akram Bayat, Marc Pomplun, and Duc A. Tran, "A study on human activity recognition using accelerometer data from smartphones," vol. 34, pp. 450–457, 12 2014.

[38] S. Amendola, A. Palombi, L. Rousseau, G. Lissorgues, and G. Marrocco, "Manufacturing technologies for uhf rfid epidermal antennas," in *2016 46th European Microwave Conference (EuMC)*, Oct 2016, pp. 914–917.

[39] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C. Leung, "Body area networks: A survey," *Mob. Netw. Appl.*, vol. 16, no. 2, pp. 171–193, Apr. 2011.

[40] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester, "A survey on wireless body area networks," *Wirel. Netw.*, vol. 17, no. 1, pp. 1–18, Jan. 2011.

[41] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, July 2013.

[42] M. Mozer, "The adaptive house," in *The IEE Seminar on Intelligent Building Environments, 2005. (Ref. No. 2005/11059)*, June 2005, pp. 39–79.

[43] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen, "The gator tech smart house: A programmable pervasive space," *Computer*, vol. 38, no. 3, pp. 50–60, Mar. 2005.

[44] Tatsuya Yamazaki, "The ubiquitous home," *International Journal of Smart Home*, 2007.

[45] Stephen S. Intille, "The goal: smart people, not smart homes," in *In Proceedings of the International Conference on Smart Homes and Health Telematics, IOS*. 2006, Press.

[46] Mukhtiar Memon, Stefan Rahr Wagner, Christian Fischer Pedersen, Femina Hassan Aysha Beevi, and Finn Overgaard Hansen, "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes," *Sensors*, vol. 14, no. 3, pp. 4312–4341, 2014.

[47] M.A. Estudillo-Valderrama, L.M. Roa, J. Reina-Tosina, and I. Román-Martínez, "Ambient assisted living: A methodological approach," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Aug 2010, pp. 2155–2158.

[48] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5-6, pp. 907–928, 1995.

[49] Willem Nico Borst and W.N. Borst, *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*, Ph.D. thesis, 9 1997.

[50] Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung, "Ontology based context modeling and reasoning using owl," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Washington, DC, USA, 2004, PERCOMW '04, pp. 18–, IEEE Computer Society.

[51] Lin Liu and Eric Yu, “Designing information systems in social context: A goal and scenario modelling approach,” *Inf. Syst.*, vol. 29, no. 2, pp. 187–203, Apr. 2004.

[52] Eric S. K. Yu, “Towards modeling and reasoning support for early-phase requirements engineering,” in *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*, Washington, DC, USA, 1997, RE ’97, pp. 226–235, IEEE Computer Society.

[53] ITU-T, “User requirements notation (urn) - language definition,” Recommendation Z.151, International Telecommunication Union, Geneva, Oct. "2012".

[54] L. Rossi, A. Belli, A. De Santis, C. Diamantini, E. Frontoni, E. Gambi, L. Palma, L. Pernini, P. Pierleoni, D. Potena, L. Raffaeli, S. Spinsante, P. Zingaretti, D. Cacciagrano, F. Corradini, R. Culmone, F. De Angelis, E. Merelli, and B. Re, “Interoperability issues among smart home technological frameworks,” 2014, cited By 1.

[55] M.J. Darlington and S.J Culley, “Investigating ontology development for engineering design support,” *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 112 – 134, 2008, Intelligent computing in engineering and architecture.

[56] G. Schreiber, B. Wielinga, R. de Hoog, H. Akkermans, and W. Van de Velde, “CommonKADS: a comprehensive methodology for KBS development,” *IEEE Expert*, vol. 9, no. 6, pp. 28–37, 1994.

[57] M. Uschold and M. Gruninger, “Ontologies: Principles, methods and applications,” *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–155, June 1996.

[58] M. Fernandez, A. Perez, and N. Juristo, “METHONTOLOGY: from Ontological Art towards Ontological Engineering,” in *Proc. of the AAAI97 Spring Symposium Series on Ontological Engineering*, Stanford, USA, March 1997, pp. 33–40.

[59] Natalya F Noy and Deborah L McGuinness, “Ontology development 101: A guide to creating your first ontology,” *Development*, vol. 32, no. 1, pp. 1–25, 2001.

[60] Karen Najera, Anna Perini, Alicia Martínez, and Hugo Estrada, “Supporting *i** model integration through an ontology-based approach,” in *Proceedings of the 5th International* i* *Workshop 2011, Trento, Italy, August 28-29, 2011*, 2011, pp. 43–48.

[61] Dario Bonino and Fulvio Corno, "DogOnt - ontology modeling for intelligent domotic environments," in *Proceedings of the 7th International Conference on The Semantic Web*, Berlin, Heidelberg, 2008, ISWC '08, pp. 790–803, Springer-Verlag.

[62] Claudia Diamantini, Domenico Potena, and Emanuele Storti, "SemPI: A semantic framework for the collaborative construction and maintenance of a shared dictionary of performance indicators," *Future Generation Computer Systems*, 2015.

[63] Compton, M. et al., "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, no. 0, 2012.

[64] Deborah L. McGuinness and Frank van Harmelen, "Owl web ontology language overview," Tech. Rep., W3C - World Wide Web Consortium, January 2004.

[65] Frank Van Harmelen and Ian Horrocks, "Reference description of the daml+oil ontology markup language: March 2001," 2001.

[66] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler, "Owl 2: The next step for owl," *Web Semant.*, vol. 6, no. 4, pp. 309–322, Nov. 2008.

[67] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, New York, NY, USA, 2003.

[68] Leon Sterling, Alan Bundy, Lawrence Byrd, Richard O'Keefe, and Bernard Silver, "Solving symbolic equations with PRESS," *J. Symb. Comput.*, vol. 7, no. 1, pp. 71–84, Jan. 1989.

[69] Ahmed Nait Aicha, Gwenn Englebienne, and Ben Kröse, *Continuous Gait Velocity Analysis Using Ambient Sensors in a Smart Home*, pp. 219–235, Springer International Publishing, Cham, 2015.

[70] Lars Donath, Oliver Faude, Eric Lichtenstein, Geert Pagenstert, Corina Nüesch, and Annegret Mündermann, "Mobile inertial sensor based gait analysis: Validity and reliability of spatiotemporal gait characteristics in healthy seniors," vol. 49, 07 2016.

[71] Jürgen Nehmer, Martin Becker, Arthur Karshmer, and Rosemarie Lamm, "Living assistance systems: An ambient intelligence approach," in *Proceedings of the 28th International Conference on Software Engineering*, New York, NY, USA, 2006, ICSE '06, pp. 43–50, ACM.

[72] Michael Klein, Andreas Schmidt, and Rolf Lauer, "Ontology-centred design of an ambient middleware for assisted living: The case of SOPRANO," in *Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU), 30th Annual German Conference on Artificial Intelligence (KI 2007), Osnabrück*, Thomas Kirste, Birgitta König-Ries, and Ralf Salomon, Eds., 2007.

[73] Peter Wolf, Andreas Schmidt, and Michael Klein, "SOPRANO - an extensible, open AAL platform for elderly people based on semantical contracts," in *3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI'08), 18th European Conference on Artificial Intelligence (ECAI 08), Patras, Greece*, 2008.

[74] Patrice Roy, Sylvain Giroux, Bruno Bouchard, Abdenour Bouzouane, Clifton Phua, Andrei Tolstikov, and Jit Biswas, "A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients," vol. 4, pp. 33–58, 04 2011.

[75] Toshiyo Tamura, Atsushi Kawarada, Masayuki Nambu, Akira Tsukada, Kazuo Sasaki, and Ken-Ichi Yamakoshi, "E-healthcare at an experimental welfare techno house in japan.," *The open medical informatics journal*, vol. 1, pp. 1–7, 2007.

[76] Jennifer Horkoff and Eric Yu, "Analyzing goal models: Different approaches and how to choose among them," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, New York, NY, USA, 2011, SAC '11, pp. 675–682, ACM.

[77] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Science of Computer Programming*, vol. 20, no. 1- 2, pp. 3 – 50, 1993.

[78] H. Kaiya, H. Horai, and M. Saeki, "AGORA: attributed goal-oriented requirements analysis method," in *Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on*, 2002, pp. 13–22.

[79] John Mylopoulos, Lawrence Chung, and Eric Yu, "From object-oriented to goal-oriented requirements analysis," *Commun. ACM*, vol. 42, no. 1, pp. 31–37, Jan. 1999.

[80] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, "TROPOS: An agent-oriented software development methodology," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, May 2004.

[81] Harry Chen, Tim Finin, and Anupam Joshi, "The soupa ontology for pervasive computing," in *Ontologies for Agents: Theory and Experiences*, Valentina Tamma, Stephen Cranefield, TimothyW. Finin, and Steven Willmott, Eds., Whitestein Series in Software Agent Technologies, pp. 233–258. Birkhäuser Basel, 2005.

[82] Lorenzo Sommaruga, Antonio Perri, and Francesco Furfari, "DomoML-env: an ontology for human home interaction," in *SWAP*, Paolo Bouquet and Giovanni Tummarello, Eds. 2005, vol. 166 of *CEUR Workshop Proceedings*, CEUR-WS.org.

[83] X.H. Wang, D.Q. Zhang, T. Gu, and H.K. Pung, "Ontology based context modeling and reasoning using OWL," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, March 2004, pp. 18– 22.

[84] Daqing Zhang, Tao Gu, and Xiaohang Wang, "Enabling context-aware smart home with semantic technology," *International Journal of Human-friendly Welfare Robotic Systems*, pp. 12–20, 2005.

[85] Giovanni Meroni and Pierluigi Plebani, *Artifact-Driven Monitoring for Human-Centric Business Processes with Smart Devices: Assessment and Improvement*, pp. 160–176, Springer International Publishing, Cham, 2017.

[86] Rachit AGARWAL, David Gomez Fernandez, Tarek Elsaleh, Amelie Gyrard, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny, "Unified IoT Ontology to Enable Interoperability and Federation of Testbeds," in *3rd IEEE World Forum on Internet of Things*, Reston, United States, Dec. 2016.

[87] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud, "Cross-domain internet of things application development: M3 framework and evaluation," in *Proceedings of the 2015 3rd International Conference on Future Internet of Things and Cloud*, Washington, DC, USA, 2015, FICLOUD '15, pp. 9–16, IEEE Computer Society.

[88] Guotong Xie, Yang Yang, Shengping Liu, Zhaoming Qiu, Yue Pan, and Xiongzhi Zhou, "EIAW: Towards a Business-Friendly Data Warehouse Using Semantic Web Technologies," in *The Semantic Web*, vol. 4825 of *Lecture Notes in Computer Science*, pp. 857–870. Springer Berlin Heidelberg, 2007.

[89] Matthias Kehlenbeck and Michael H. Breitner, "Ontology-based exchange and immediate application of business calculation definitions for

online analytical processing," in *Proc. of the 11th International Conference on Data Warehousing and Knowledge Discovery*, Berlin, Heidelberg, 2009, DaWaK '09, pp. 298–311, Springer-Verlag.

[90] M. Golfarelli, F. Mandreoli, W. Penzo, S. Rizzi, and E. Turricchia, "OLAP Query Reformulation in Peer-to-peer Data Warehousing," *Information Systems*, vol. 37, no. 5, pp. 393–411, July 2012.

[91] Carlos Pedrinaci and John Domingue, "Ontology-based metrics computation for business process analysis," in *Proc. of the 4th International Workshop on Semantic Business Process Management*, 2009, pp. 43–50.

[92] Viara Popova and Viara Sharpanskykh, "Formal modelling of organisational goals based on performance indicators," *Data & Knowledge Engineering*, vol. 70, no. 4, pp. 335–364, 2011.

[93] Alireza Pourshahid, Gregory Richards, and Daniel Amyot, "Toward a goal-oriented, business intelligence decision-making framework.," in *MCETECH*, Gilbert Babin, Katarina Stanoevska-Slabeva, and Peter Kropf, Eds. 2011, vol. 78 of *Lecture Notes in Business Information Processing*, pp. 100–115, Springer.

[94] Alex Rashidi, Parisa; Mihailidis, "A survey on ambient-assisted living tools for older adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, 2013.

[95] Luigi Atzori, Antonio Iera, and Giacomo Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[96] Wil M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer Publishing Company, Incorporated, 1st edition, 2011.

[97] Wil van der Aalst, Ton Weijters, and Laura Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 9, pp. 1128–1142, September 2004.

[98] Sander J. J. Leemans, Dirk Fahland, and Wil M. P. van der Aalst, "Discovering block-structured process models from event logs containing infrequent behaviour," in *Business Process Management Workshops*, Beijing, China, Aug, 26 2013, pp. 66–78.

[99] Josep Carmona, Jordi Cortadella, and Michael Kishinevsky, "Genet: A tool for the synthesis and mining of petri nets," in *Proceedings of the 2009 Ninth International Conference on Application of Concurrency to*

System Design, Washington, DC, USA, 2009, ACSD '09, pp. 181–185, IEEE Computer Society.

[100] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik, *Process Discovery Using Integer Linear Programming*, pp. 368–387, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[101] Wil M.P. van der Aalst, *Process Mining in the Large: A Tutorial*, pp. 33–76, Springer International Publishing, Cham, 2014.

[102] Dirk Fahland and Wil M. P. van der Aalst, *Repairing Process Models to Reflect Reality*, pp. 229–245, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[103] Wil M. P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer Publishing Company, Incorporated, 1st edition, 2011.

[104] Wil M. P. Günther, Christian W.and van der Aalst, *Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics*, pp. 328–343, Springer Berlin Heidelberg, 2007.

[105] A. Weijters, W.M.P. van der Aalst, and A.K.A. de Medeiros, "Process mining with the heuristics miner-algorithm," *Technische Universiteit Eindhoven, Tech. Rep. WP*, vol. 166, 2006.

[106] Claudia Diamantini, Laura Genga, Domenico Potena, and Wil M. P. van der Aalst, "Building instance graphs for highly variable processes," *Expert Syst. Appl.*, vol. 59, pp. 101–118, 2016.

[107] S Katz, A B Ford, R W Moskowitz, B A Jackson, and M W Jaffe, "Studies of illness in the aged. the index of adl: a standardized measure of biological and psychosocial function.," *JAMA*, vol. 185, pp. 914–9, 1963.

[108] S Katz, "Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living.," *Journal of the American Geriatrics Society*, vol. 31, no. 12, pp. 721–7, 1983.

[109] 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*, vol. 3001 of *LNCS*, pp. 158–175. Springer Berlin Heidelberg, 2004.

[110] Parisa Rashidi and Diane J. Cook, "Mining and monitoring patterns of daily routines for assisted living in real world settings," in *Proceedings of the 1st ACM International Health Informatics Symposium*, New York, NY, USA, 2010, pp. 336–345, ACM.

[111] Prafulla N Dawadi, Diane J Cook, Maureen Schmitter-Edgecombe, and Carolyn Parsey, "Automated assessment of cognitive health using smart home technologies.," *Technology and health care : official journal of the European Society for Engineering and Medicine*, vol. 21, no. 4, pp. 323–43, 2013.

[112] Shamila Nasreen, Muhammad Awais Azam, Usman Naeem, Mustansar Ali Ghazanfar, and Asra Khalid, "Recognition framework for inferring activities of daily living based on pattern mining," *Arabian Journal for Science and Engineering*, vol. 41, no. 8, pp. 3113–3126, Aug 2016.

[113] Tâm Huynh, Mario Fritz, and Bernt Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, New York, NY, USA, 2008, UbiComp '08, pp. 10–19, ACM.

[114] Timo Sztyler, Johanna Völker, Josep Carmona, Oliver Meier, and Heiner Stuckenschmidt, "Discovery of personal processes from labeled sensor data - an application of process mining to personalized health care," in *Proceedings of the International Workshop on Algorithms and Theories for the Analysis of Event Data*, 2015, pp. 31–46.

[115] Carlos Fernández-Llatas, Aroa Lizondo, Eduardo Monton Sanchez, José-Miguel Benedí, and Vicente Traver, "Process mining methodology for health process tracking using real-time indoor location systems," *Sensors*, vol. 15, no. 12, pp. 29821–29840, 2015.

[116] Arik Senderovich, Andreas Rogge-Solti, Avigdor Gal, Jan Mendling, and Avishai Mandelbaum, "The road from sensor data to process instances via interaction mining," in *Int Conf on Advanced Information Systems Engineering*, June, 13-17 2016, pp. 257–273.

[117] Marcella Dimaggio, Francesco Leotta, Massimo Mecella, and Daniele Sora, "Process-based habit mining: Experiments and techniques," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, July 18-21 2016, pp. 145–152.

[118] Zakwan Jaroucheh, Xiaodong Liu, and Sally Smith, "Recognize contextual situation in pervasive environments using process mining techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, pp. 53–69, 2011.

[119] Carlos Fernandez-Llatas, Bernardo Valdivieso, Vicente Traver, and Jose Miguel Benedi, *Using Process Mining for Automatic Support of Clinical Pathways Design*, pp. 79–88, Springer New York, 2015.

[120] C. Fernandez-Llatas, S. F. Pileggi, V. Traver, and J. M. Benedi, "Timed parallel automaton: A mathematical tool for defining highly expressive formal workflows," in *2011 Fifth Asia Modelling Symposium*, May 2011, pp. 56–61.

[121] Simona D'Oca, Stefano Corgnati, and Tianzhen Hong, "Data mining of occupant behavior in office buildings," *Energy Procedia*, vol. 78, no. Supplement C, pp. 585 – 590, 2015, 6th International Building Physics Conference, IBPC 2015.

[122] Ryan Shaun Baker and Paul Salvador Inventado, *Educational Data Mining and Learning Analytics*, pp. 61–75, Springer New York, New York, NY, 2014.

[123] Commissione Paritetica per la didattica e il diritto allo studio, "Relazione Annuale 2016, Facoltà di Ingegneria, Università Politecnica delle Marche," December 2016.

[124] C. Diamantini, L. Genga, D. Potena, and E. Storti, "Pattern discovery from innovation processes," in *2013 Int Conf on Collaboration Technologies and Systems*, May 2013, pp. 457–464.