future internet

MDPI

Review

Surveying Human Habit Modeling and Mining Techniques in Smart Spaces

Francesco Leotta *, Massimo Mecella *, Daniele Sora and Tiziana Catarci

Dipartimento di Ingegneria Informatica, Automatica e Gestionale Antonio Ruberti, Sapienza Università di Roma, 00185 Rome, Italy; sora@diag.uniroma1.it (D.S.); catarci@diag.uniroma1.it (T.C.)

* Correspondence: leotta@diag.uniroma1.it (F.L.); mecella@diag.uniroma1.it (M.M.)

Received: 28 December 2018; Accepted: 16 January 2019; Published: 19 January 2019

check for updates

Abstract: A smart space is an environment, mainly equipped with Internet-of-Things (IoT) technologies, able to provide services to humans, helping them to perform daily tasks by monitoring the space and autonomously executing actions, giving suggestions and sending alarms. Approaches suggested in the literature may differ in terms of required facilities, possible applications, amount of human intervention required, ability to support multiple users at the same time adapting to changing needs. In this paper, we propose a Systematic Literature Review (SLR) that classifies most influential approaches in the area of smart spaces according to a set of dimensions identified by answering a set of research questions. These dimensions allow to choose a specific method or approach according to available sensors, amount of labeled data, need for visual analysis, requirements in terms of enactment and decision-making on the environment. Additionally, the paper identifies a set of challenges to be addressed by future research in the field.

Keywords: smart spaces; intelligent environments; survey; human habits; modeling

1. Introduction

The progress of information and communication technologies has many faces; while computing speed, reliability and level of miniaturization of electronic devices increase year after year, their costs decrease. This allows a widespread adoption of *embedded systems* (e.g., appliances, sensors, actuators) and of powerful computing devices (e.g., laptops, smartphones), thus, turning *pervasive* (or *ubiquitous*) computing into reality. Pervasive computing embodies a vision of computers seamlessly integrating into everyday life, responding to information provided by sensors in the environment, with little or no direct instruction from users [1]. At the same time, connecting all these computing devices together, as networked artefacts, using local and global network infrastructures has become easy. The rise of applications that exploit these technologies represents a major characteristic of the *Internet-of-Things* (IoT) [2].

Smart spaces represent an emerging class of IoT-based applications. Smart homes and offices are representative examples where pervasive computing could take advantage of *ambient intelligence* (AmI) more easily than in other scenarios where artificial intelligence—AI problems soon become intractable [3].

A study about the current level of adoption of commercial smart home systems is provided in [4]. This study reveals how people understanding of the term "smart" has a more general meaning than what we presented here as AmI; in particular, it also includes non-technological aspects such as the spatial layout of the house. Additionally, an automated behavior is considered as smart, especially from people without a technical background, only if it performs a task quicker than the user could do by himself. The research also reveals that interest in smart home systems is subject to a virtuous circle such that people experiencing benefits from their services feel the need of upgrading them.

The universAAL specification [5] defines a smart space as "an environment centered on its human users in which a set of embedded networked artefacts, both hardware and software, collectively realize the paradigm of *ambient intelligence*" (AmI).

Many different definitions of AmI are provided in the literature, e.g., [6] introduces a set of distinguishing keywords characterizing AmI systems, namely: sensitivity, responsiveness, adaptivity, ubiquity and transparency. The term *sensitivity* refers to the ability of an AmI system to sense the environment and, more generally, to understand the *context* it is interacting with. Strictly related to sensitivity are *responsiveness* and *adaptivity*, which denote the capability to timely act, in a reactive or proactive manner, in response to changes in the context according to user preferences (*personalization*). Sensitivity, responsiveness and adaptivity all contribute to the concept of *context awareness*. Finally, the terms *ubiquity* and *transparency* directly refer to concept of pervasive computing. AmI is obtained by merging techniques from different research areas [5] including artificial intelligence (AI) and human-computer interaction (HCI).

The different proposed approaches largely differs in terms of models taken as input for both space dynamics and human behavior in it, how these models are extracted, what kind of input sensors they need, which are the constraints under which they are supposed to effectively work.

The performance evaluation of all these approaches, especially in the academic field, is often conducted in controlled situations where some of the features of a real environment are hid by the need for repeatability of the experiment.

In this paper, we propose a comparative framework for techniques and approaches for modeling and extracting models to be employed with modern smart spaces. Differently from other surveys in the same area, this work focuses on giving an overview of the approaches by enforcing a taxonomy with dimensions that are chosen in order to understand the suitability of a specific technique to a specific setting.

The paper is organized as it follows. Section 2 introduces the terminology and basic notions of smart spaces needed to fruitfully follow the discussion. Section 3 explains the inclusion criteria for the literature review. Section 4 discusses sensor types and how they are considered in the relevant literature. Sections 5 and 6 discuss the different types of smart space models employed and how they are constructed. Section 7 compares our work with other similar literature surveys. Finally, Section 8 concludes the paper by providing final considerations.

2. Background

Figure 1 depicts the closed loops that characterize a running smart space [7]. The main closed loop, depicted using solid arrows and shapes, shows how the *knowledge* of environment dynamics and of users behaviors and preferences is employed to interpret *sensors* output in order to perform appropriate *actions* on the environment. Sensor data is first analyzed to extract the current *context*, which is an internal abstraction of the state of the environment from the point of view of the AmI system. The extracted context is then employed to *make decisions* on the actions to perform on the controlled space. Actions related to these decisions modify the environment (both physical and digital) by means of *actuators* of different forms.

Sensors can be roughly divided into *physical* ones, which provide direct measurements about the environment (e.g., humidity, brightness, temperature), the devices and the users, and *cyber* ones, which provide digital information, not directly related to physical phenomena, such as user calendars. A cyber sensor often provides information related to the presence of the user in the cyberspace (e.g., the calendar of an user, a tweet posted by him/her, etc.).

The term *sensor data* encompasses raw (or minimally processed) data retrieved from both physical sensors and cyber sensors. We can imagine a smart space producing, at runtime, a *sensor log* containing raw measurements from available sensors.

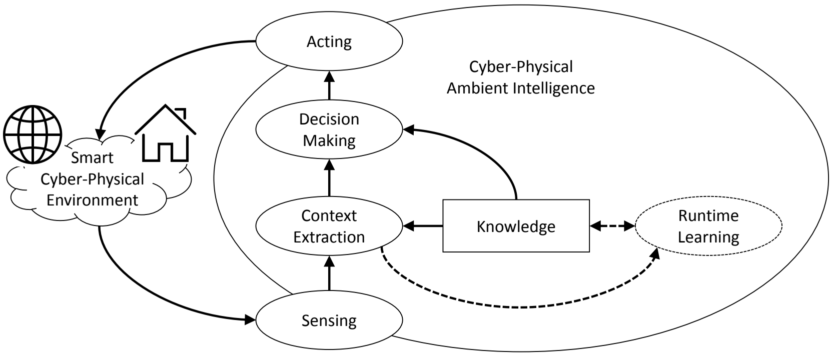


Figure 1. The ambient intelligence closed loop. Arrows denote information flow. Dashed lines are used to denote optional information flows.

Definition 1 (Sensor Log)**.** *Given a set S of sensors, a sensor log is a sequence of measurements of the kind $\langle ts, s, v\rangle$ where ts is the timestamp of the measurement, $s \in S$ is the source sensor and v the measured value, which can be either nominal (categorical) or numeric (quantitative).*

Measurements can be produced by a sensor on a periodic base (e.g., temperature measurements) or whenever a particular event happens (e.g., door openings). As many of the algorithms proposed in the literature borrow the terminology of data mining, the sensor log could be conceived as a sequence of events instead of a sequence of measurements.

Definition 2 (Event Log)**.** *Given a set $E = \{e_1, \ldots, e_{n_E}\}$ of event types, an event sequence is a sequence of pairs $\langle e, t\rangle$, where $e \in E$ and t is an integer, the occurrence time of the event type e.*

Definition 2 is more restrictive than Definition 1. Translating a sensor log into an event log could cause a loss of information especially if discretization of periodic sensor measurements is required.

Authors in the field of smart spaces uses, sometimes as synonyms, a variety of terms to refer to the state of the environment and the tasks humans perform in it. For the rest of the article, we will use the following terminology:

- **Context:** The state of the environment including the human inhabitants. This includes the output of sensors and actuators, but also the state of human inhabitants including the action/activities/habits he/she is performing. In this very comprehensive meaning, the term *situation* is sometimes used.
- **Action:** Atomic interaction with the environment or a part of it (e.g., a device). Recognizing actions can be easy or difficult depending on the sensors installed. Certain methods only focuses on actions and they will not be part of our survey. In some cases methods to recognize activities and habits completely skip the action recognition phase, only relying on the raw measurements in the sensor log.
- **Activity:** A sequence of actions (one in the extreme case) or sensor measurements/events with a final goal. In some cases an action can be an activity itself (e.g., ironing). Activities can be collaborative, including actions by multiple users and can interleave one each other. The granularity (i.e., the temporal extension and complexity) of considered activities cannot be precisely specified. According to the approach, tidying up a room can be an activity whereas others approaches may generically consider tidying up the entire house as an activity. In any case, some approaches may hierarchically define activities, where an activity is a combination of sub-activities.
- **Habit:** A sequence or interleaving of activities that happen in specific contextual conditions (e.g., what the user does every morning between 08:00 and 10:00).

Knowledge plays a central role in AmI systems. As it intervenes both for context extraction and decision-making, it takes the form of a set of models describing (i) users behavior,

(ii) environment/device dynamics, and (iii) user preferences. Anyway, knowledge should not be considered as a static resource as both users behavior and preferences change over time. Vast majority of works in the area of ambient intelligence suppose the knowledge to be obtained off-line, independently from the system runtime. A second optional loop in Figure 1, depicted using dashed arrows, shows that the current context could be employed to update the knowledge by applying learning techniques at runtime.

Noteworthy, AmI is not intended to be provided by a centralized entity, on the contrary, its nature is distributed with embedded devices and software modules, possibly unaware one of each other, contributing to its features. Recently introduced smart space appliances, such as the NEST thermostat (see https://nest.com/thermostats/nest-learning-thermostat), contain sensors, actuators and AmI features in a single small package.

3. Inclusion Criteria and Comparison Framework

In order to conduct our analysis we took inspiration from SLR (Systematic Literature Review) guidelines. A SLR is a method to identify, evaluate and interpret relevant scientific works with respect to a specific topic. We designed a protocol for conducting the SLR inspired to the guidelines and policies presented in [8], in order to ensure that the results are repeatable and the means of knowledge acquisition are scientific and transparent.

The necessary steps to guarantee compliance with the guidelines include (i) the formulation of the research questions; (ii) the definition of a search string; (iii) the selection of the data sources on which the search is performed; (iv) the identification of inclusion and exclusion criteria; (v) the selection of studies; (vi) the method of extracting data from the studies; and (vii) the analysis of the data.

Our analysis covers the different aspects of habit mining by separately analyzing the features requested to the sensor log, the modeling phase, and the runtime phase. The modeling phase is in charge of creating the models of human habits and environmental dynamics, whereas the runtime phase covers the aspect related to how these models are employed at runtime to recognize the context and to act on the environment. These two phases can even overlap in the case the system is able to refine models at runtime (either in collaboration or not with the user) or even to completely create them from scratch at runtime. In any case, the runtime phase covers only the way models are employed whereas the modeling phase cover any phase that is related to model production or update.

On the basis of the above premises, the following research questions **RQ-x** have been defined:

- **RQ-A:** Sensor measurements represent the traces of user behavior in the environment. This information is needed at runtime to understand the current context, but the specific available information must be known when the model is defined (through specification or learning). The current review will only take into account the sensors for which the method has been validated. Section 4 will present for each of the included papers the following information:

  - **RQ-A1:** which sensors are taken into account?
  - **RQ-A2:** are the sensors only categorical or numerical? And in the latter case,
  - **RQ-A3:** which discretization strategy is employed?

- **RQ-B1:** Any proposed method has a way to represent models. Models can be represented using graphical or mathematical/logic formalisms. Some methods propose formalisms that are specifically designed for the particular approach. Other methods conversely employ standard formalisms from machine learning and data mining. Section 5 analyzes the following aspects about employed models:

  - **RQ-B1.1:** the type of adopted model;
  - **RQ-B1.2:** how does an instance of the model can be represented? Is it human readable?
  - **RQ-B1.3:** which is the granularity at which the model works?

- **RQ-B2:** Whichever model type is adopted by a specific method, each method introduces a different way to construct the model. Section 6 expands the following analysis:

- **RQ-B2.1:** for methods involving an, optionally partial, automatic construction of models (i.e., through learning), a training set consisting of a sensor log must be fed as input. At first, learning methods in the field of ambient intelligence can be classified according to the effort devoted to label the training set;
- **RQ-B2.2:** does the system consider the possibility of having multiple users? How many?
- **RQ-B2.3:** in the latter case, is some type of additional labeling needed?

As terms in this specific research area are not yet standardized, we replaced the search string with a crawling-like procedure starting from a set of seed papers, recursively navigating papers through the "Cited by" feature of Google Scholar, and selecting influential works from the last 10 years. This kind of search was manually performed on Google Scholar, as it includes all the relevant sources of scientific papers available.

We then computed the 33rd percentile over the number of citations per year. In general, using the number of citations per year promotes recent works, as they attract citations more than outdated ones, but as we are analyzing only works from the last 10 years, this approach allows to highlight the most influential works. Additionally, papers that extend works included by this criterion are included in the results.

The application of this criteria allowed to identify 22 *primary studies* (cf. Table 1) that were included in the final SLR. For each work, in the following tables, a general description is provided in addition to some schematic information.

Table 1. Selected studies.

Acronym	**Study Title**
AUG-ECA [9]	Augusto, J. C., Liu, J., McCullagh, P., Wang, H., and Yang, J.-B. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home.
AUG-APUBS [10]	Aztiria, A., Augusto, J. C., Basagoiti, R., Izaguirre, A., and Cook, D. J. Discovering frequent user–environment interactions in intelligent environments.
CASAS-DISCOREC [11–13]	Cook, D. J., Krishnan, N. C., and Rashidi, P. Activity discovery and activity recognition: A new partnership.
	Rashidi, P., Cook, D. J., Holder, L. B., and Schmitter-Edgecombe, M. Discovering activities to recognize and track in a smart environment.
	Rashidi, P. and Cook, D. J. Com: A method for mining and monitoring human activity patterns in home-based health monitoring systems.
CASAS-HMM [14]	Singla, G., Cook, D. J., and Schmitter-Edgecombe, M. Recognizing independent and joint activities among multiple residents in smart environments.
CASAS-HAM [15]	Rashidi, P. and Cook, D. J. Keeping the resident in the loop: Adapting the smart home to the user.
CASAS-HMMNBCRF [16]	Cook, D. J. Learning setting-generalized activity models for smart spaces.
CASAS-SVM [17]	Krishnan, N. C. and Cook, D. J. Activity recognition on streaming sensor data.
CHEN-ONT [18]	Chen, L., Nugent, C. D., and Wang, H. A knowledge-driven approach to activity recognition in smart homes.
WANG-EP [19]	Gu, T., Wang, L., Wu, Z., Tao, X., and Lu, J. A pattern mining approach to sensor-based human activity recognition.

Table 1. *Cont.*

Acronym	Study Title
RIB-PROB [20,21]	Helaoui, R., Riboni, D., and Stuckenschmidt, H. A probabilistic ontological framework for the recognition of multilevel human activities.
	Riboni, D., Sztyler, T., Civitarese, G., and Stuckenschmidt, H. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning.
NUG-EVFUS [22]	Hong, X., Nugent, C., Mulvenna, M., McClean, S., Scotney, B., and Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes.
WANG-HIER [23]	William, C. et al. Fast effective rule induction.
KROS-CRF [24]	Van Kasteren, T., Noulas, A., Englebienne, G., and Kröse, B. Accurate activity recognition in a home setting.
REIG-SITUATION [25]	Brdiczka, O., Crowley, J. L., and Reignier, P. Learning situation models in a smart home.
YANG-NN [26]	Yang, J.-Y., Wang, J.-S., and Chen, Y.-P. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers.
STIK-MISVM [27]	Stikic, M., Larlus, D., Ebert, S., and Schiele, B. Weakly supervised recognition of daily life activities with wearable sensors.
FLEURY-MCSVM [28]	Fleury, A., Vacher, M., and Noury, N. Svm-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results.
PALMES-OBJREL [29]	Palmes, P., Pung, H. K., Gu, T., Xue, W., and Chen, S. Object relevance weight pattern mining for activity recognition and segmentation.
MOORE-STRAWMAN [30]	Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers.
LES-PHI [31]	Lester, J., Choudhury, T., and Borriello, G. A practical approach to recognizing physical activities.
BUE-WISPS [32]	Buettner, M., Prasad, R., Philipose, M., and Wetherall, D. Recognizing daily activities with rfid-based sensors.

4. Supported Sensors

Sensing technologies have made significant progress on designing sensors with smaller size, lighter weight, lower cost, and longer battery life. Sensors can, thus, be embedded in an environment and integrated into everyday objects and onto human bodies without affecting users' comfort. Nowadays, sensors do not only include those traditionally employed for home and building automation (e.g., presence detectors, smoke detectors, contact switches for doors and windows, network-attached and close circuit cameras) but also more modern units (e.g., IMU—Inertial Measurements Units such as accelerometer and gyroscopes, WSN nodes), which are growingly available as part of off-the-shelf products.

Measured values are usually affected by a certain degree of uncertainty. Sensors have indeed their own technical limitations as they are prone to breakdowns, disconnections from the system and environmental noise (e.g., electromagnetic noise). As a consequence, measured values can be out of date, incomplete, imprecise, and contradictory with each other. Techniques for cleaning sensor data do exist [33], but uncertainty of sensor data may still lead to wrong conclusions about the current context, which in turn potentially lead to incorrect behaviors of the system.

Formalisms employed for representing knowledge in AmI systems often need environmental variables to be *binary* or *categorical*. A wide category of sensors (e.g., temperature sensors) produce instead numerical values, making it necessary to discretize sensor data before they can be used for reasoning.

Discretization methods in machine learning and data mining are usually classified according to the following dimensions [34]:

- *Supervised* vs. *Unsupervised*. Unsupervised methods do not make use of class information in order to select cut-points. Classic unsupervised methods are equal-width and equal-frequency binning, and clustering. Supervised methods employ instead class labels in order to improve discretization results.
- *Static* vs. *Dynamic*. Static discretization methods perform discretization, as a preprocessing step, prior to the execution of the learning/mining task. Dynamic methods instead carry out discretization on the fly.
- *Global* vs. *Local*. Global methods, such as binning, are applied to the entire n-dimensional space. Local methods, as the C4.5 classifier, produce partitions that are applied to localized regions of the instance space. A local method is usually associated with a dynamic discretization method.
- *Top-down* vs. *Bottom-up*. Top-down methods start with an empty list of cut-points (or split-points) and keep on adding new ones to the list by splitting intervals as the discretization progresses. Bottom-up methods start with the complete list of all the continuous values of the feature as cut-points and remove some of them by merging intervals as the discretization progresses.
- *Direct* vs. *Incremental*. Direct methods directly divide the range of a quantitative attribute in k intervals, where the parameter k is provided as input by the user. Conversely, incremental methods start from a simple discretization and improve it step by step in order to find the best value of k.

An important aspect to take into account when evaluating the usage of a particular category of sensors is the final user acceptance. Wearable sensors or cameras usually provoke a certain level of embarrassment in the users that must be taken into account.

Table 2 shows, for each of the selected papers, the answers to research questions A1, A2 and A3. A3 is always empty for methods only supporting discrete sensors.

Table 2. Supported Sensors—RQ-A1: sensors type—RQ-A2: data produced type—RQ-A3: discretization strategy.

Approach	RQ-A1	RQ-A2	RQ-A3
AUG-ECA [9]	Position	Discrete	-
AUG-APUBS [10]	Object sensor, motion sensor, and context sensor	Mixed	-
CASAS-DISCOREC [11–13]	motion + interaction (RFID)	Discrete	-
CASAS-HMM [14]	motion + interaction (RFID)	Discrete	-
CASAS-HMMNBCRF [16]	motion + interaction (RFID)	Discrete	-
CASAS-HAM [15]	motion + interaction(RFID)	Discrete	-
CASAS-SVM [17]	Passive InfraRed	Discrete	-
CHEN-ONT [18]	Dense sensed environment: RFID, PIR, context sensors (tilt, pressure)	Mixed	Sensor measurements continuously fed into the system.
RIB-PROB [20,21]	RFID + body sensor	Discrete	-
WANG-EP [19]	RFID + user's movement, environmental information, user location, human-object interaction	Mixed	From analogical measurements, like acceleration data and temperature, statistical values are computed (e.g., mean, variance)
NUG-EVFUS [22]	PIR, contact, pressure mats	Discrete	-
WANG-HIER [23]	RFID + body sensors (IMOTE2)	Discrete	-
KROS-CRF [24]	large variety of sensors, from contact to humidity/temperature	Mixed	Temporal probabilistic model: sampling measurements creating sequences of observations

Table 2. *Cont.*

Approach	RQ-A1	RQ-A2	RQ-A3
REIG-SITUATION [25]	Video and audio records	Analogical	Extraction of user position and speed ($|\vec{v}|$)
YANG-NN [26]	triaxial accelerometer on a wearable board	Discrete	-
STIK-MISVM [27]	Audio-video, object interaction, environmental conditions, appliances' use, 3D accelerometers	Mixed	Statistical features computed over 30 s sliding window shifted in increments of 15 s.
FLEURY-MCSVM [28]	Many: position, microphones, contact, wearable	Mixed	Extraction of statistical features (e.g., number of events per microphone)
PALMES-OBJREL [29]	RFID	Discrete	-
LES-PHI [31]	Accelerometers + audio + barometric pressure	Mixed	
BUE-WISPS [32]	WISPS: UHF RFID + accelerometers	Discrete	-

5. Model Types

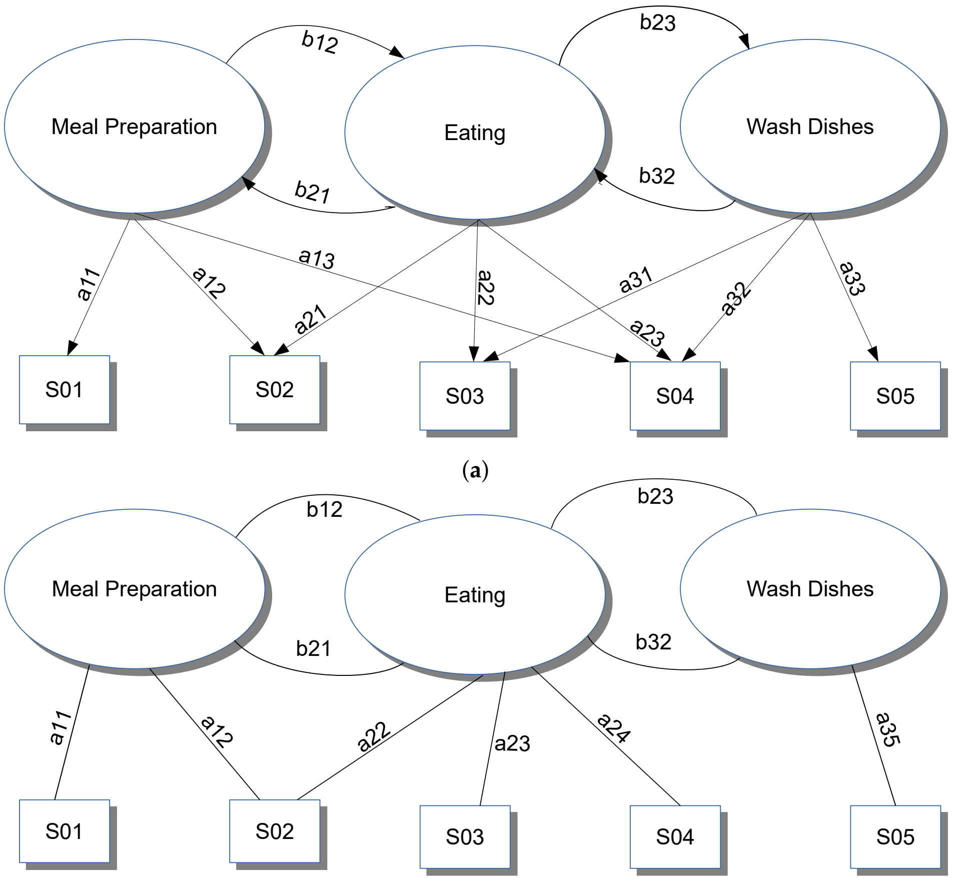
Knowledge is represented in AmI systems using models. The literature about representing models of human habits is wide. In this section, we will review the most adopted approaches, highlighting those formalisms that are human understandable, thus, being easy to validate by a human expert or by the final user (once the formalism is known).

Bayesian classification techniques are based on the well known Bayes theorem $P(H|X) = \frac{P(X|H)P(H)}{P(X)}$, where H denotes the hypothesis (e.g., a certain activity is happening) and X represents the set of evidences (i.e., the current value of context objects). As calculating $P(X|H)$ can be very expensive, different assumptions can be made to simplify the computation. For example, *naïve Bayes* (NB) is a simple classification model, which supposes the n single evidences composing X independent (that the occurrence of one does not affect the probability of the others) given the situational hypothesis; this assumption can be formalized as $P(X|H) = \prod_{k=1}^{n} P(x_k|H)$. The inference process within the naïve Bayes assumption chooses the situation with the maximum a posteriori (MAP) probability.

Hidden Markov Models (HMMs) represent one of the most widely adopted formalism to model the transitions between different states of the environment or humans. Here hidden states represent situations and/or activities to be recognized, whereas observable states represent sensor measurements. HMMs are a statistical model where a system being modeled is assumed to be a Markov chain, which is a sequence of events. A HMM is composed of a finite set of hidden states (e.g., s_{t-1}, s_t, and s_{t+1}) and observations (e.g., o_{t-1}, o_t, and o_{t+1}) that are generated from states. HMM is built on three assumptions: (i) each state depends only on its immediate predecessor; (ii) each observation variable only depends on the current state; and (iii) observations are independent from each other. In a HMM, there are three types of probability distributions: (i) prior probabilities over initial state $p(s_0)$; (ii) state transition probabilities $p(s_t|s_{t-1})$; and (iii) observation emission probabilities $p(o_t|s_t)$.

A drawback of using a standard HMM is its lack of hierarchical modeling for representing human activities. To deal with this issue, several other HMM alternatives have been proposed, such as *hierarchical* and *abstract* HMMs. In a hierarchical HMM, each of the hidden states can be considered as an autonomous probabilistic model on its own; that is, each hidden state is also a hierarchical HMM.

HMMs generally assume that all observations are independent, which could possibly miss long-term trends and complex relationships. *Conditional Random Fields*—CRFs, on the other hand, eliminate the independence assumptions by modeling the conditional probability of a particular sequence of hypothesis, Y, given a sequence of observations, X; succinctly, CRFs model $P(Y|X)$. Modeling the conditional probability of the label sequence rather than the joint probability of both the labels and observations $P(X, Y)$, as done by HMMs, allows CRFs to incorporate complex features of the observation sequence X without violating the independence assumptions of the model. The graphical model representations of a HMM (a directed graph, Figure 2a) and a CRF (an undirected graph, Figure 2b) makes this difference explicit. In [24], a comparison between HMM and CRF is shown, where CRF outperforms HMM in terms of timeslice accuracy, while HMM outperforms CRF in terms of class accuracy.



(**a**)

(**b**)

Figure 2. Examples of HMM and CRF models. Ellipses represent states (i.e., activities). Rectangles represent sensors. Arrows between states are state transition probabilities (i.e., the probability of moving from a state to another), whereas those from states to sensors are emission probabilities (i.e., the probability that in a specific state a sensor has a specific value). (**a**) HMM model example. Picture inspired by CASAS-HMM [14] and CASAS-HAM [15]. (**b**) CRF model example. Picture inspired by KROS-CRF [24].

Another statistical tool often employed is represented by Markov Chains (MCs), which are based on the assumption that the probability of an event is only conditional to the previous event. Even if they are very effective for some applications like capacity planning, in the smart spaces context, they are quite limited because they deal with deterministic transactions and modeling an intelligent environment with this formalism results in a very complicated model.

Support Vector Machines (SVMs) allow to classify both linear and non-linear data. A SVM uses a non-linear mapping to transform the original training data into an higher dimension. Within this new dimension, it searches for the linear optimal separating hyperplane that separates the training data of one class from another. With an appropriate non-linear mapping to a sufficiently high dimension, data from two classes can always be separated. SVMs are good at handling large feature spaces since they employ overfitting protection, which does not necessarily depend on the number of features. Binary Classifiers are built to distinguish activities. Due to their characteristics, SVMs are better in generating other kind of models with a machine learning approach than modeling directly the smart environment. For instance in [35] authors uses them combined with Naive Bayes Classifiers to learn the activity model built on hierarchical taxonomy formalism shown in Figure 3.

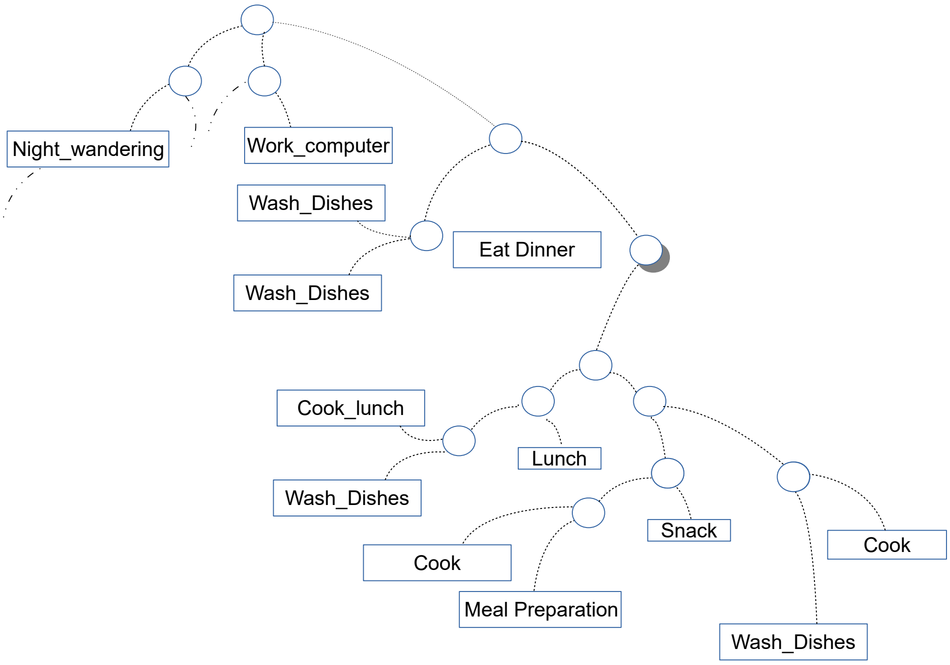


Figure 3. Cluster hierarchical model portion example obtained by exploiting SVMs. Picture inspired by [35].

Artificial Neural Networks (ANNs) are a sub-symbolic technique, originally inspired by biological neuron networks. They can automatically learn complex mappings and extract a non-linear combination of features. A neural network is composed of many artificial neurons that are linked together according to a specific network architecture. A neural classifier consists of an input layer, a hidden layer, and an output layer. Mappings between input and output features are represented in the composition of activation functions f at a hidden layer, which can be learned through a training process performed using gradient descent optimization methods or resilient backprogagation algorithms.

Some techniques stem from data mining methods for *market basket analysis* (e.g., the Apriori algorithm [36]), which apply a windowing mechanism in order to transform the event/sensor log into what is called a *database of transactions*. Let $I = \{i_1, \ldots, i_{n_E}\}$ be a set of binary variables corresponding to sensor event types. A *transaction* is an assignment that binds a value to each of the variables in I, where the values 0 and 1 respectively denote the fact that a certain event happened or not during the considered window. A *database of transactions* $\mathcal{T}$ is an (usually ordered) sequence of transactions each having a, possibly empty, set of properties (e.g., a timestamp). An *item* is an assignment of the kind $i_k = \{0, 1\}$. An *itemset* is an assignment covering a proper subset of the variables in I. An itemset C has support $Supp_{\mathcal{T}}(C)$ in the database of transactions $\mathcal{T}$ if a fraction of $Supp_{\mathcal{T}}(C)$ of transactions in the database contain C. The techniques following this strategy turn the input log into a database of transactions, each of them corresponding to a window. Given two different databases of transactions T_1 and T_2, the *growth rate* of an itemset C from T_1 to T_2 is defined as $\frac{Supp_{T_2}(X)}{Supp_{T_1}(X)}$. *Emerging patterns* (EP) are those itemsets showing a growth rate greater than a certain threshold ρ. The ratio behind this definition is that an itemset that has high support in its target class (database) and low support in the contrasting class, can be seen as a strong signal, in order to discover the class of a test instance containing it. Market basket analysis is a special case of *affinity analysis* that discovers co-occurrence relationships among purchased items inside a single or more transactions.

Initial approaches to the development of context-aware systems able to recognize situations were based on *predicate logic*. Loke [37] introduced a PROLOG extension called LogicCAP; here the "in-situation" operator captures a common form of reasoning in context-aware applications, which is to ask if an entity *E* is in a given situation *S* (denoted as $S^* > E$). In particular, a situation is defined as a set of constraints imposed on output or readings that can be returned by sensors, i.e., if *S* is the current situation, we expect the sensors to return values satisfying some constraints associated with *S*. LogicCAP rules use backward chaining like PROLOG, but also utilizes forward chaining in determining situations, i.e., a mix of backward and forward chaining is used in evaluating LogicCAP programs. The work introduces different reasoning techniques with situations including selecting the best action to perform in a certain situation, understanding what situation a certain entity is in (or the most likely) and defining relationships between situations.

There are many approaches borrowed from information technology areas, adapted to smart environments. For instance in [38], the authors use temporal logic and model checking to perform activities modeling and recognition. The system proposed is called ARA. A graphical representation of a model example adopted by this approach is shown in Figure 4. It evidences how the activities are composed by the time correlated states between consecutive actions.

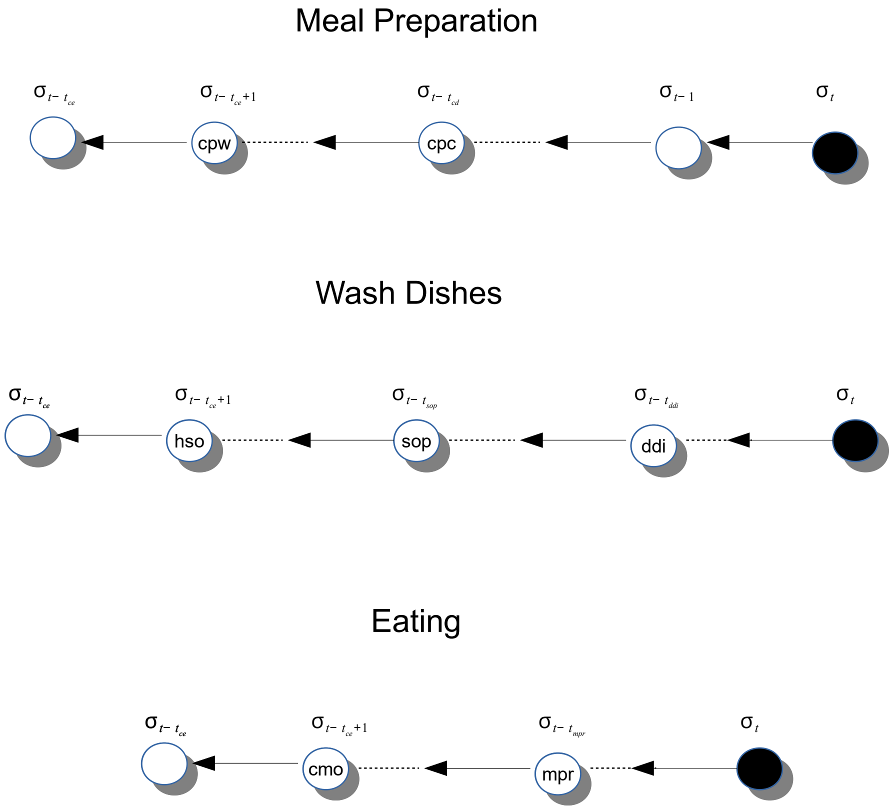
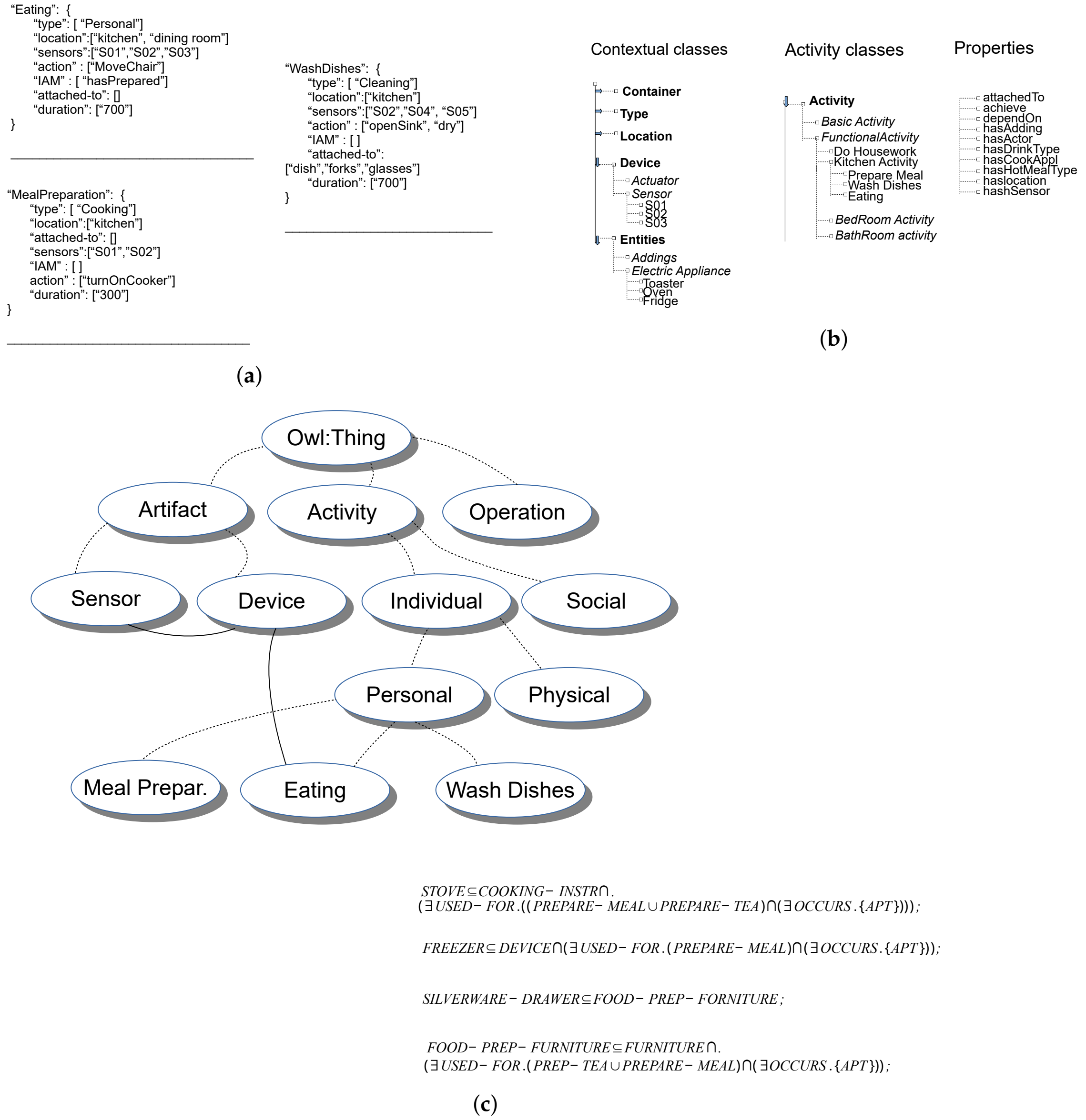


Figure 4. Model formalism used in [38] based on R-pTL formulas. The model is based on correlations between events.

Ontologies (denoted as ONTO) represent the last evolution of logic-based approaches and have increasingly gained attention as a generic, formal and explicit way to "capture and specify the domain knowledge with its intrinsic semantics through consensual terminology and formal axioms and constraints" [39]. They provide a formal way to represent sensor data, context, and situations into well-structured terminologies, which make them understandable, shareable, and reusable by both humans and machines. A considerable amount of knowledge engineering effort is expected in constructing the knowledge base, while the inference is well supported by mature algorithms and rule engines. Some examples of using ontologies in identifying situations are given by [40] (later evolved in [20,21]). Instead of using ontologies to infer activities, they use ontologies to validate the result inferred from statistical techniques.

Figure 5 contains three different examples of ontologies employed to model smart spaces. Clearly, each approach adopts the formalism most devised for its application, and many aspects can be modeled under different granularities.



```
“Eating”:  {
      “type”: [ “Personal”]
      “location”:[“kitchen”, “dining room”]
      “sensors”:[“S01”,”S02”,”S03”]
      “action” : [“MoveChair”]
      “IAM” : [ “hasPrepared”]
      “attached-to”: []
      “duration”: [“700”]
}
```

```
“MealPreparation”:  {
      “type”: [ “Cooking”]
      “location”:[“kitchen”]
      “attached-to”: []
      “sensors”:[“S01”,”S02”]
      “IAM” : [ ]
      action” : [“turnOnCooker”]
      “duration”: [“300”]
}
```

```
“WashDishes”:  {
      “type”: [ “Cleaning”]
      “location”:[“kitchen”]
      “sensors”:[”S02”,”S04”, “S05”]
      “action” : [“openSink”, “dry”]
      “IAM” : [ ]
      “attached-to”:
[“dish”,”forks”,”glasses”]
      “duration”: [“700”]
}
```

(**a**)

(**b**)

$$STOVE \subseteq COOKING-INSTR \cap .$$
$$(\exists USED-FOR.((PREPARE-MEAL \cup PREPARE-TEA) \cap (\exists OCCURS.\{APT\})));$$

$$FREEZER \subseteq DEVICE \cap (\exists USED-FOR.(PREPARE-MEAL) \cap (\exists OCCURS.\{APT\}));$$

$$SILVERWARE-DRAWER \subseteq FOOD-PREP-FORNITURE;$$

$$FOOD-PREP-FURNITURE \subseteq FURNITURE \cap .$$
$$(\exists USED-FOR.(PREP-TEA \cup PREPARE-MEAL) \cap (\exists OCCURS.\{APT\}));$$

(**c**)

Figure 5. Examples of Ontology Models. (**a**) Example of machine-oriented ontology. Picture taken from [41]. (**b**) Example of taxonomy based ontology. Picture inspired by CHEN-ONT [18]. (**c**) A portion of the resulting ontology model proposed in RIB-PROB [20,21]. The dashed lines represent the relations super/sub between classes.

The way an AmI system makes decisions on the actions can be compared to decision-making in AI agents. As an example, *reflex agents with state*, as introduced in [42], take as input the current state of the world and a set of Condition-Action rules to choose the action to be performed. Similarly, Augusto [43] introduces the concept of Active DataBase (ADB) composed by Event-Condition-Action (ECA) rules. An ECA rule basically has the form “ON *event* IF *condition* THEN *action*”, where conditions can take into account time.

The first attempts to apply techniques taken from the business process management—BPM [44] area were the employment of workflow specifications to anticipate user actions. A workflow is composed by a set of tasks related by qualitative and/or quantitative time relationships. Authors in [45] present a survey of techniques for *temporal calculus* (i.e., Allen’s Temporal Logic and Point

Algebra) and *spatial calculus* aiming at decision-making. The SPUBS system [46,47] automatically retrieves these workflows from sensor data.

Table 3 shows, for each surveyed paper, information about RQ-B1.1 (Formalism), RQ-B1.2 (Readability Level) and RQ-B1.3 (Granularity).

Table 3. Model type (L—Low, M—Medium, H—High)—RQ-B1.1: The type of model—RQ-B1.2: Human Readability Level—RQ-B1.3: Granularity Level.

References	RQ-B1.1	RQ-B1.2	RQ-B1.3
AUG-ECA [9]	ECA	H	Action
AUG-APUBS [10]			Action
CASAS-HMMNBCRF [16]	NB	L	Activity
CASAS-DISCOREC [11–13]	HMM	M	Activity
CASAS-HMM [14]			Activity
CASAS-HMMNBCRF [16]			Activity
KROS-CRF [24]			Activity
REIG-SITUATION [25]			Situation
LES-PHI [31]			Activity
BUE-WISPS [32]			Activity
CASAS-HAM [15]	MC	M	Event
CASAS-HMMNBCRF [16]	CRF	L	Activity
KROS-CRF [24]			Activity
REIG-SITUATION [25]	SVM	L	Situation
STIK-MISVM [27]			Activity
FLEURY-MCSVM [28]			Activity
CHEN-ONT [18]	ONTO	H	Activity
RIB-PROB [20,21]			Action/Activity
NUG-EVFUS [22]			Action
WANG-EP [19]	EP	L	Action/Activity
YANG-NN [26]	ANN	L	Activity
PALMES-OBJREL [29]	Other	H	Activity

6. Model Construction

Modeling formalisms in the literature can be roughly divided into *specification-based* and *learning-based* [1]. Research in the field of AmI started when few kinds of sensors were available and the relationships between sensor data and underlying phenomena were easy to establish. Specification-based approaches represent hand-made *expert* knowledge in logic rules and apply reasoning engines to infer conclusions and to make decisions from sensor data. These techniques evolved in the last years in order to take into account uncertainty. The growing availability of different kinds of sensors made hand-made models impractical to be produced. In order to solve this problem, learning-based methods employ techniques from machine learning and data mining. Specification-based models are usually more *human-readable* (even though a basic experience with formal logic languages is required), but creating them is very expensive in terms of human resources. Most learning-based models are instead represented using mathematical and statistical formalisms (e.g., HMMs), which make them difficult to be revised by experts and understood by final users. These motivations are at the basis of the research of *human-readable automatically inferred formalisms*.

Learning-based techniques can be divided into *supervised* and *unsupervised* techniques. The former expect the input to be previously labeled according to the required output function, hence, they require a big effort for organizing input data in terms of training examples, even though active learning can be employed to ease this task. Unsupervised techniques (or weakly supervised ones, i.e., those where only a part of the dataset is labeled) can be used to face this challenge but a limited number of works is available in the literature.

Unsupervised techniques for AmI knowledge modeling can be useful for other two reasons. Firstly, as stated in the introduction, sometimes knowledge should not be considered as a static resource; instead it should be updated at runtime without a direct intervention of the users [15], hence, updating techniques should rely on labeling of sensor data as little as possible. Moreover, unsupervised techniques may also result useful in supporting *passive users*, such as guests, that do not participate in the configuration of the system but should benefit from its services as well.

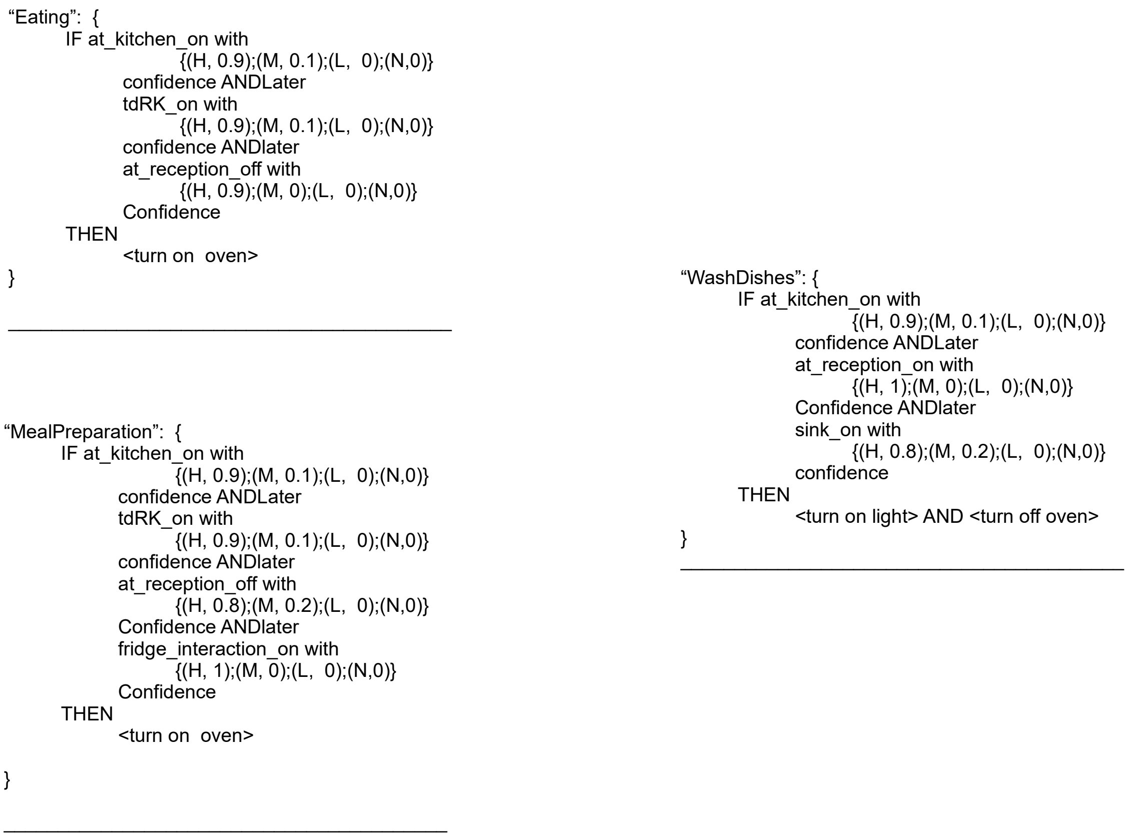
Performing learning or mining from sequences of sensor measurements poses the issue of how to group events into aggregates of interests (i.e., actions, activities, situations). Even with supervised learning techniques, if labeling is provided at learning time, the same does not hold at runtime where a stream of events is fed into the AmI system. Even though most proposed approaches in the AmI literature (especially supervised learning ones) ignore this aspect, windowing mechanism are needed. As described in [17], the different windowing methods can be classified into three main classes, namely, *explicit*, *time-based* and *event-based*.

- *Explicit segmentation.* In this case, the stream is divided into chunks usually following some kind of classifier previously instructed over a training data set. Unfortunately, as the training data set simply cannot cover all the possible combinations of sensor events, the performance of such a kind of approach usually results in single activities divided into multiple chunks and multiple activities merged.
- *Time-based windowing.* This approach divides the entire sequence into equal size time intervals. This is a good approach when dealing with data obtained from sources (e.g., sensors like accelerometers and gyroscopes) that operate continuously in time. As can be easily argued, the choice of the window size is fundamental especially in the case of sporadic sensors as a small window size could not contain enough information to be useful whereas a large window size could merge multiple activities when burst of sensors occur.
- *Sensor Event-based windowing.* This last approach splits the entire sequence into bins containing an equal number of sensor events. Usually, bins are in this case overlapping with each window containing the last event arrived together with the previous events. Whereas this method usually performs better than the other, it shows drawbacks similar to those introduced for time based windowing.

In AUG-ECA [9], the rules composing the model are written by experts, so the model formalism is based on knowledge bases. The rules will be evaluated by exploiting evidential reasoning (ER) and the parameter for the ER are trained in a supervised way. The rules structure follows the Event-Condition-Action (ECA) paradigm, borrowed by database techniques. The rules specify how the system has to react to a given event in a specific context. The action performed by the system influences the status of the system itself, potentially generating another event managed with a reaction is other rules.

Figure 6 shows some examples of ECA rules used to model some activities. The structure has the typical layout of a IF-THEN construct, in which the conditions represent sensor triggering (or its interpretation), and the conditions are expressed as probabilities regarding the context elements and the auxiliary actions performed. The limits of this approach are related to the big effort needed to model successfully the environment and the possibility of producing conflicting conditions.

The evolution in specification-based approaches to overcome the previous limits is represented by ontologies. In CHEN-ONT [18], an activity is modeled as a class: the entities and the properties correlated to them are divided in three groups (see Figure 7). The first one is about the context, the second group represents the causal and/or functional relations, the properties of the third group denote the type and interrelationship between activities.



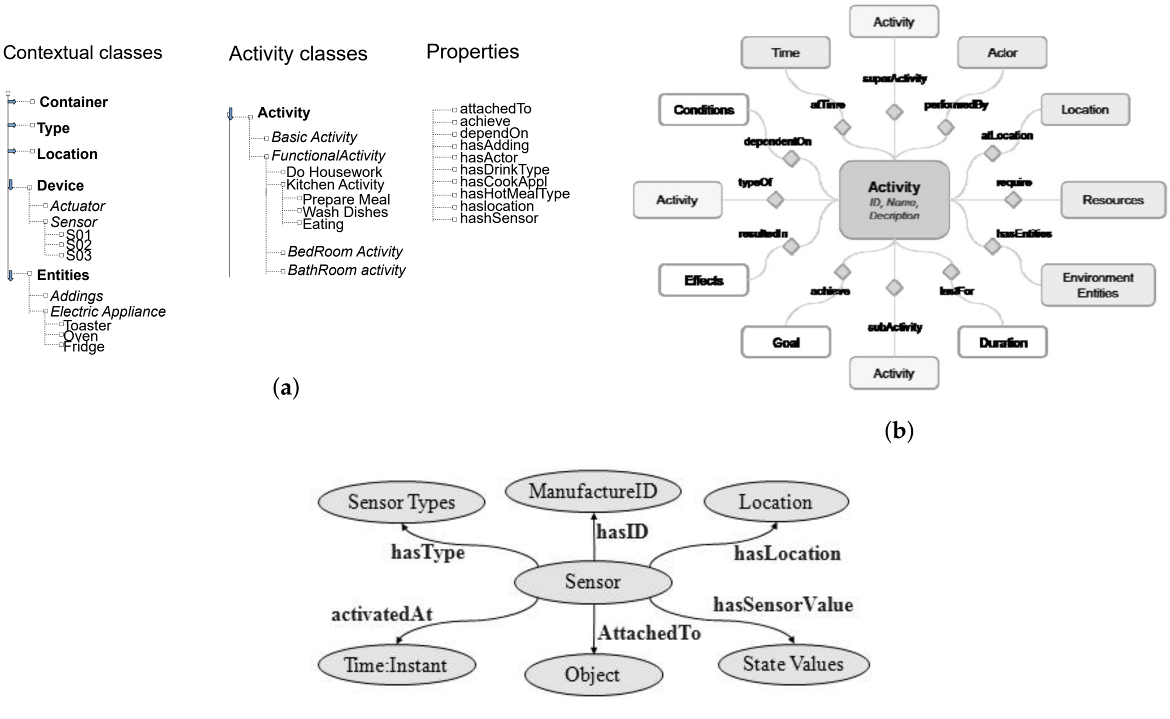
```
"Eating": {
        IF at_kitchen_on with
                    {(H, 0.9);(M, 0.1);(L,  0);(N,0)}
               confidence ANDLater
               tdRK_on with
                    {(H, 0.9);(M, 0.1);(L,  0);(N,0)}
               confidence ANDlater
               at_reception_off with
                    {(H, 0.9);(M, 0);(L,  0);(N,0)}
               Confidence
        THEN
               <turn on  oven>
}
```

```
"MealPreparation": {
        IF at_kitchen_on with
                    {(H, 0.9);(M, 0.1);(L,  0);(N,0)}
               confidence ANDLater
               tdRK_on with
                    {(H, 0.9);(M, 0.1);(L,  0);(N,0)}
               confidence ANDlater
               at_reception_off with
                    {(H, 0.8);(M, 0.2);(L,  0);(N,0)}
               Confidence ANDlater
               fridge_interaction_on with
                    {(H, 1);(M, 0);(L,  0);(N,0)}
               Confidence
        THEN
               <turn on  oven>

}
```

```
"WashDishes": {
        IF at_kitchen_on with
                    {(H, 0.9);(M, 0.1);(L,  0);(N,0)}
               confidence ANDLater
               at_reception_on with
                    {(H, 1);(M, 0);(L,  0);(N,0)}
               Confidence ANDlater
               sink_on with
                    {(H, 0.8);(M, 0.2);(L,  0);(N,0)}
               confidence
        THEN
               <turn on light> AND <turn off oven>
}
```

Figure 6. Example of ECA rules modeling "Eating", "MealPreparation", "WashDishes" activities. Pictures inspired by AUG-ECA [9].



(**a**)

(**b**)

(**c**)

Figure 7. Ontologies used in CHEN-ONT [18] to model the aspects of smart spaces. (**a**) Ontology example used to model the Smart Environment domain. Picture inspired to CHEN-ONT [18]. (**b**) Ontology example used to model the activities correlations. Picture taken from CHEN-ONT [18]. (**c**) Ontology example used to model the sensors properties. Picture taken from CHEN-ONT [18].

In RIB-PROB [20,21], the multilevel model is obtained combining ontologies and/or grouping elements at previous levels. The Atomic Gestures model is obtained just considering log elements. The Manipulative Gestures are computed considering ontology and axioms. The Simple Activities are obtained grouping Manipulative Gestures. Finally, for Complex Activities, ontologies are involved. Figure 5c represents a portion of the resulting ontology model. The dashed lines represent the relations super/sub between classes. The individual classes have relations that describe dependencies. Moreover Description Logic is employed to support ontological reasoning, which allows also to check the knowledge base consistency. It also infers additional information from registered facts.

In NUG-EVFUS [22], the interrelationships between sensors, context and activities are represented as a hierarchical network of ontologies (see Figure 8). A particular activity can be performed or associated with a certain room in the house, this information is modeled with an ontology of the network.

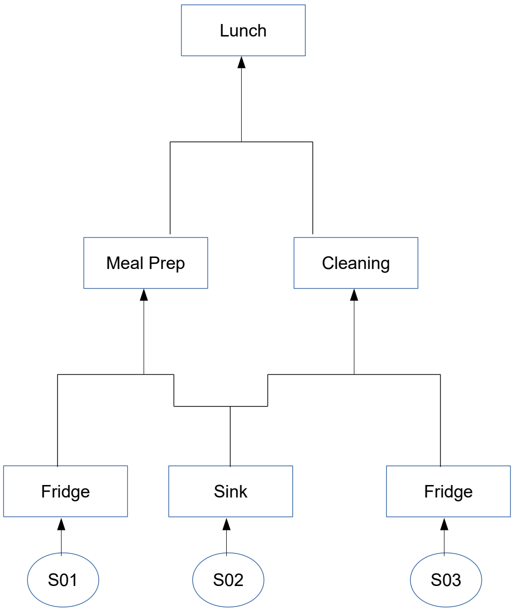


Figure 8. Hierarchical ontology structure adopted in NUG-EVFUS [22] to model activities in a smart space.

In CASAS-HMM [14], each activity is performed in a protected environment, and the resulting log is recorded and labeled. Then, HMM model is built upon this dataset in a supervised way. The resulting model is shown in Figure 2a. Observations (squares) model the sensor triggering, the states (circles) model the activities that can generate the observations according to certain probabilities. The goal is to infer the activities by processing the observations.

This recognition technique is supporting single user data, but the problem of modeling multiple users is introduced. The same team, in CASAS-SVM [17], employs SVM. In this second work, authors propose an interesting analysis of the different windowing strategies to be employed to gather measurements into observation vectors. Finally, in CASAS-HMMNBCRF [16], experiments are performed with the same methodology adding CRF and NB modeling techniques to the analysis.

In WANG-EP [19], from log of sequential activities, Emerging Patterns are mined and the resulting set composes the model.

In KROS-CRF [24], the model is trained out from a labeled dataset. The log is divided in segments 60-s long and each segment is labeled. The dataset is composed by multi-days logs: one day is used for testing the approach, the remaining for training the models. The resulting model is an undirected graph as in Figure 2b.

In REIG-SITUATION [25], a SVM model, built on statistical values extracted from the measurements of a given user, is used for classifying the roles. Then, this information, combined with other statistically extracted ones, is involved into the training of the HMM that models the situations.

In YANG-NN [26], the input vector contains the features to consider, the output vector the classes (activities). The back-propagation learning algorithm is used for training the ANNs. Three neural networks are built on labeled logs: a pre-classifier and two classifiers; static activities and dynamic activities are modeled with separated ANNs. The structure of the neural classifier consists of an input layer, an hidden layer and an output layer.

In LES-PHI [31], given the maximum number of features the activity recognition system can use, the system automatically chooses the most discriminative sub-set of features and uses them to learn an ensemble of discriminative static classifiers for the activities that need to be recognized. Then, the class probabilities estimated from the static classifiers are used as inputs into HMMs.

In BUE-WISPS [32], the users are asked to perform activities. The resulting log is used for training an HMM.

In FLEURY-MCSVM [28], the classes of the classifier model the activities. Binary classifiers are built to distinguish activities: pairwise combinations selection. The number of SVMs for n activities will be $n-1$. The features used are statistics from measurements.

The algorithm proposed in CASAS-DISCOREC [11–13] is to improve the performance of activity recognition algorithms by trying to reduce the part of the dataset that has not been labeled during data acquisition. In particular, for the unlabeled section of the log, the authors employ a pattern mining technique in order to discover, in an unsupervised manner, human activity patterns. A pattern is defined here as a set of events where order is not specified and events can be repeated multiple times. Patterns are mined by iteratively compressing the sensor log. The data mining method used for activity discovery is completely unsupervised without the need of manually segmenting the dataset or choosing windows and allows to discover interwoven activities as well. Starting from singleton patterns, at each step, the proposed technique compresses the logs by exploiting them and iteratively reprocesses the compressed log for recognizing new patterns and further compress the log. When it is difficult to further compress the log, each remaining pattern represents an activity class. Discovered labels are employed to train HMM, BN and SVM models following the same approach as in the supervised works of the same group.

In CASAS-HAM [15], the sensor log is considered completely unlabeled. Here temporal patterns (patterns with the addition of temporal information) are discovered similarly as in CASAS-DISCOREC [11–13] and are used for structuring a tree of Markov Chains. Different activations in different timestamps generate new paths in the tree. Depending to temporal constraints, a sub-tree containing Markov Chains at the leafs that model activities is generated. A technique to update the model is also proposed. Here the goal of the model is the actuation of target devices more than recognition.

Authors in STIK-MISVM [27] introduce a weakly supervised approach where two strategies are proposed for assigning labels to unlabeled data. The first strategy is based on the miSVM algorithm. miSVM is a SVM with two levels, the first for assigning labels to unlabeled data, the second one for applying recognition to the activities logs. The second strategy is instead called graph-based label propagation, where the nodes of the graphs are vectors of features. The nodes are connected by weighted edges, the weights represent similarity between nodes. When the entire training set is labeled, an SVM is trained for activity recognition.

In AUG-APUBS [10], the system generates ECA rules by considering the typology of the sensors involved in the measurements and the time relations between their activations. APUBS makes clear the difference between different categories of sensors:

- **Type O** sensors installed in objects, thus, providing direct information about the actions of the users.
- **Type C** sensors providing information about the environment (e.g., temperature, day of the week).
- **Type M** sensors providing information about the position of the user inside the house (e.g., in the bedroom).

Events in the *event* part of the ECA rule always come from sets O and M. Conditions are usually expressed in terms of the values provided by Type C sensors. Finally, the *action* part contains only Type O sensors.

The set of Type O sensor is called *mainSeT*. The first step of the APUBS method consists of discovering, for each sensor in the *mainSeT*, the set *associatedSeT* of potential O and M sensors that can be potentially related to it as triggering events. The method employed is APriori for association rules [36]; the only difference is that possible association rules $X \Rightarrow Y$ are limited to those where cardinality of both X and Y is unitary and Y only contains events contained in *mainSeT*. Obviously, this step requires a window size value to be specified in order to create transactions. As a second step, the technique discovers the temporal relationships between the events in *associatedSeT* and those in *mainSeT*. During this step, non-significant relations are pruned. As a third step, the conditions for the ECA rules are mined with a JRip classifier [48].

In WANG-HIER [23], starting from the raw log, the authors use a K-Medoids clustering method to discover template gestures. This method finds the k representative instances which best represent the clusters. Based on these templates, gestures are identified applying a template matching algorithm: Dynamic Time Warping is a classic dynamic programming based algorithm to match two time series with temporal dynamics.

In PALMES-OBJREL [29], the KeyExtract algorithm mines keys from the web that best identify activities. For each activity, the set of most important keys is mined. In the recognition phase, an unsupervised segmentation based on heuristics is performed.

Table 4 shows a quick recap of this section and answers questions RQ-B2.1 (Technique Class), RQ-B2.2 (Multi-user Support) and RQ-B2.3 (Additional Labeling Requirement).

Table 4. Model construction (Y—Yes, N—No, S—Single, M—Multiple)—RQ-B2.1: Specification vs Learning (Supervised or Unsupervised)—RQ-B2.2: Number of users—RQ-B2.3: Labeling to address multi-user.

Technique	**RQ-B2.1**	**RQ-B2.2**	**RQ-B2.3**
AUG-ECA [9]	Specification	S	N
CHEN-ONT [18]	Specification	S	N
RIB-PROB [20,21]	Specification	S	N
NUG-EVFUS [22]	Specification	S	N
CASAS-HMM [14]	Supervised	S	N
CASAS-SVM [17]	Supervised	S	N
CASAS-HMMNBCRF [16]	Supervised	S	N
WANG-EP [19]	Supervised	S	N
KROS-CRF [24]	Supervised	S	N
REIG-SITUATION [25]	Supervised	M	N
YANG-NN [26]	Supervised	S	N
LES-PHI [31]	Supervised	S	N
BUE-WISPS [32]	Supervised	S	N
FLEURY-MCSVM [28]	Supervised	M	Y
CASAS-DISCOREC [11–13]	Weakly Sup.	S	N
STIK-MISVM [27]	Weakly Sup.	M	Y
AUG-APUBS [10]	Unsupervised	S	N
CASAS-HAM [15]	Unsupervised	S	N
WANG-HIER [23]	Unsupervised	S	N
PALMES-OBJREL [29]	Unsupervised	S	N

7. Related Work

The literature contains several surveys attempting to classify works in the field of smart spaces and ambient intelligence. Papers are presented in chronological orders. None of the reported surveys, clearly states the modality by which papers have been selected.

Authors in [7] follow an approach similar to this work, i.e., they separately analyze the different phases of the life-cycle of the models. Differently from our work, for what concerns the model construction phase, they focus on classes of learning algorithms instead of analyzing the specific work. Additionally, specification-based methods are not taken into account.

The survey [49] is focused on logical formalisms to represent ambient intelligence contexts and reasoning about them. The analyzed approaches are solely specification-based. Differently from our work, the survey is focused on the reasoning aspect, which is not the focus of our work.

The work [50] is an extensive analysis of methods employed in ambient intelligence. This work separately analyzes the different methods without clearly defining a taxonomy.

Authors in [1] introduce a clear taxonomy of approaches in the field of context recognition (and more generally, about situation identification). The survey embraces the vast majority of the proposed approaches in the area. Equivalently, paper [51] is a complete work covering not only activity recognition but also fine-grained action recognition. Differently from our work, both surveys are not focusing on the life-cycle of models.

Authors in [52] focus on reviewing the possible applications of ambient intelligence in the specific case of health and elderly care. The work is orthogonal to the present paper and all the other reported works, as it is less focused on the pros and cons of each approach, while instead focusing on applications and future perspectives.

A manifesto of the applications and principles behind smart spaces and ambient intelligence is presented in [53].

As in [52], authors in [54] take moves from the health care application scenario in order to describe possible applications. Anyway, this work goes more into details of employed techniques with particular focus on classical machine learning methods.

8. Discussion

In this paper, we applied the SLR approach to classify most prominent approaches in the ambient intelligence area according to how they approach the different stages of a model life cycle. The kind of classification introduced in the paper allows to choose a specific method or approach according to available sensors, amount of labeled data, need for visual analysis, requirements in terms of enactment and decision-making on the environment.

The field of ambient intelligence has been very active in the last 10 years, and first products (such as NEST) have hit the market. Unfortunately, as of 2018, no examples of holistic approaches, such as those proposed in the vast majority of academic works, are available outside research labs. The reasons for this are multiple.

First of all, the vast majority of approaches either require a big effort in terms of *(i)* expert time to produce hand-made models in specification based approaches, or *(ii)* user time to manually label training sets for supervised learning based approaches. Unsupervised methods represent a small percentage on the total amount of academic works.

Secondarily, model update is often not taken into account by the proposed approaches. Human habits, the way of performing activities and preferences generally change over time. In the vast majority of cases, proposed solutions do not provide an explicit method to update models, thus, requiring modeling to start from scratch, with the drawbacks identified in the first step.

As a third point, support for multiple users performing actions separately and multiple users collaborating in a single activity are often neglected by proposed approaches. Supporting multiple users poses serious limitations to the applicability of available solutions. This problem is even harder to solve when human guests must be taken into account.

As a final point, the validity of the declared results is often difficult to confirm on datasets different from the one used for tests, as in the vast majority of cases source code is not made available. The lack of benchmark data, even though many datasets are made freely available by research groups, makes this situation even harder.

Author Contributions: F.L. contributed to methodology, conceptualization, writing and editing; M.M. contributed to methodology, writing and editing, supervision and funding acquisition; D.S. contributed to investigation, conceptualization, writing and editing; T.C. contributed to supervision and funding acquisition.

Funding: The work of Daniele Sora has been partly supported by the Lazio regional project *SAPERI & Co* (FILAS-RU-2014-1113), the work of Francesco Leotta has been partly supported by the Lazio regional project *Sapientia* (FILAS-RU-2014-1186), all the authors have been also partly supported by the Italian projects *Social Museum & Smart Tourism* (CTN01-00034-23154), *NEPTIS* (PON03PE-00214-3) and *RoMA—Resilence of Metropolitan Areas* (SCN-00064).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAL	Ambient Assisted Living
ADB	Active DataBase
AI	Artificial Intelligence
AmI	Ambient Intelligence
(A)NN	(Artificial) Neural Network
BPM	Business Process Management
CRF	Conditional Random Field
DST	Dampster-Shafer Theory
ECA	Event-Condition-Action
EP	Emerging Pattern
HCI	Human-Computer Interaction
HMM	Hidden Markov Model
IMU	Inertial Measurements Unit
IoT	Internet-of-Things
MC	Markov Chain
NB	Naive Bayes
RFID	Radio-Frequency IDentification
RQ	Research Question
SLR	Systematic Literature Review
SVM	Support Vector Machine
UHF	Ultra-High Frequency
WSN	Wireless Sensor Network

References

1. Ye, J.; Dobson, S.; McKeever, S. Situation identification techniques in pervasive computing: A review. *Pervasive Mob. Comput.* **2012**, *8*, 36–66. [CrossRef]
2. Uckelmann, D.; Harrison, M.; Michahelles, F. An architectural approach towards the future Internet-of-Things. In *Architecting the Internet-of-Things*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–24.
3. Augusto, J.; Nugent, C. (Eds.) Smart Homes Can Be Smarter. In *Designing Smart Homes*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4008, pp. 1–15.
4. Mennicken, S.; Huang, E. Hacking the Natural Habitat: An In-the-Wild Study of Smart Homes, Their Development, and the People Who Live in Them. In *Pervasive Computing*; Lecture Notes in Computer Science; Kay, J., Lukowicz, P., Tokuda, H., Olivier, P., Krüger, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7319, pp. 143–160.

5. Tazari, M.R.; Furfari, F.; Fides-Valero, Á.; Hanke, S.; Höftberger, O.; Kehagias, D.; Mosmondor, M.; Wichert, R.; Wolf, P. The universAAL Reference Model for AAL. In *Handbook of Ambient Assisted Living*; IOS Press Ebooks: Amsterdam, The Netherlands, 2012; Volume 11, pp. 610–625.
6. Cook, D.; Augusto, J.; Jakkula, V. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive Mob. Comput.* **2009**, *5*, 277–298. [CrossRef]
7. Aztiria, A.; Izaguirre, A.; Augusto, J.C. Learning patterns in ambient intelligence environments: A survey. *Artif. Intell. Rev.* **2010**, *34*, 35–51. [CrossRef]
8. Kitchenham, B. *Procedures for Performing Systematic Reviews*; Keele University: Keele, UK, 2004; Volume 33, pp. 1–26.
9. Augusto, J.C.; Liu, J.; McCullagh, P.; Wang, H.; Yang, J.B. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. *Int. J. Comput. Intell. Syst.* **2008**, *1*, 361–378. [CrossRef]
10. Aztiria, A.; Augusto, J.C.; Basagoiti, R.; Izaguirre, A.; Cook, D.J. Discovering frequent user–environment interactions in intelligent environments. *Pers. Ubiquitous Comput.* **2012**, *16*, 91–103. [CrossRef]
11. Cook, D.J.; Krishnan, N.C.; Rashidi, P. Activity discovery and activity recognition: A new partnership. *IEEE Trans. Cybern.* **2013**, *43*, 820–828. [CrossRef] [PubMed]
12. Rashidi, P.; Cook, D.J.; Holder, L.B.; Schmitter-Edgecombe, M. Discovering activities to recognize and track in a smart environment. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 527–539. [CrossRef]
13. Rashidi, P.; Cook, D.J. COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems. *ACM Trans. Intell. Syst. Technol.* **2013**, *4*, 64. [CrossRef]
14. Singla, G.; Cook, D.J.; Schmitter-Edgecombe, M. Recognizing independent and joint activities among multiple residents in smart environments. *J. Ambient Intell. Human. Comput.* **2010**, *1*, 57–63. [CrossRef]
15. Rashidi, P.; Cook, D.J. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2009**, *39*, 949–959. [CrossRef]
16. Cook, D.J. Learning setting-generalized activity models for smart spaces. *IEEE Intell. Syst.* **2012**, *27*, 32–38. [CrossRef] [PubMed]
17. Krishnan, N.C.; Cook, D.J. Activity recognition on streaming sensor data. *Pervasive Mob. Comput.* **2014**, *10*, 138–154. [CrossRef]
18. Chen, L.; Nugent, C.D.; Wang, H. A knowledge-driven approach to activity recognition in smart homes. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 961–974. [CrossRef]
19. Gu, T.; Wang, L.; Wu, Z.; Tao, X.; Lu, J. A pattern mining approach to sensor-based human activity recognition. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 1359–1372. [CrossRef]
20. Helaoui, R.; Riboni, D.; Stuckenschmidt, H. A probabilistic ontological framework for the recognition of multilevel human activities. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; ACM: New York, NY, USA, 2013; pp. 345–354.
21. Riboni, D.; Sztyler, T.; Civitarese, G.; Stuckenschmidt, H. Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; ACM: New York, NY, USA, 2016; pp. 1–12.
22. Hong, X.; Nugent, C.; Mulvenna, M.; McClean, S.; Scotney, B.; Devlin, S. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive Mob. Comput.* **2009**, *5*, 236–252. [CrossRef]
23. Wang, L.; Gu, T.; Tao, X.; Lu, J. A hierarchical approach to real-time activity recognition in body sensor networks. *Pervasive Mob. Comput.* **2012**, *8*, 115–130. [CrossRef]
24. Van Kasteren, T.; Noulas, A.; Englebienne, G.; Kröse, B. Accurate activity recognition in a home setting. In Proceedings of the 10th International Conference on Ubiquitous computing, Seoul, Korea, 21–24 September 2008; ACM: New York, NY, USA, 2008; pp. 1–9.
25. Brdiczka, O.; Crowley, J.L.; Reignier, P. Learning situation models in a smart home. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2009**, *39*, 56–63. [CrossRef]
26. Yang, J.Y.; Wang, J.S.; Chen, Y.P. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recognit. Lett.* **2008**, *29*, 2213–2220. [CrossRef]
27. Stikic, M.; Larlus, D.; Ebert, S.; Schiele, B. Weakly supervised recognition of daily life activities with wearable sensors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2521–2537. [CrossRef]

28. Fleury, A.; Vacher, M.; Noury, N. SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 274–283. [CrossRef]
29. Palmes, P.; Pung, H.K.; Gu, T.; Xue, W.; Chen, S. Object relevance weight pattern mining for activity recognition and segmentation. *Pervasive Mob. Comput.* **2010**, *6*, 43–57. [CrossRef]
30. Kwapisz, J.R.; Weiss, G.M.; Moore, S.A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explor. Newsl.* **2011**, *12*, 74–82. [CrossRef]
31. Lester, J.; Choudhury, T.; Borriello, G. A practical approach to recognizing physical activities. In Proceedings of the International Conference on Pervasive Computing, Dublin, Ireland, 7–10 May 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–16.
32. Buettner, M.; Prasad, R.; Philipose, M.; Wetherall, D. Recognizing daily activities with RFID-based sensors. In Proceedings of the 11th International Conference on Ubiquitous Computing, Orlando, FL, USA, 30 September–3 October 2009; ACM: New York, NY, USA, 2009; pp. 51–60.
33. Aggarwal, C.C. *Managing and Mining Sensor Data*; Springer Publishing Company, Incorporated: New York, NY, USA, 2013.
34. Liu, H.; Hussain, F.; Tan, C.L.; Dash, M. Discretization: An enabling technique. *Data Min. Knowl. Discov.* **2002**, *6*, 393–423. [CrossRef]
35. Krishnan, N.; Cook, D.J.; Wemlinger, Z. Learning a taxonomy of predefined and discovered activity patterns. *J. Ambient Intell. Smart Environ.* **2013**, *5*, 621–637. [PubMed]
36. Agrawal, R.; Srikant, R. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago de Chile, Chile, 12–15 September 1994; Volume 1215, pp. 487–499.
37. Loke, S.W. Logic programming for context-aware pervasive computing: Language support, characterizing situations, and integration with the web. In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China, 20–24 September 2004; pp. 44–50.
38. Magherini, T.; Fantechi, A.; Nugent, C.D.; Vicario, E. Using temporal logic and model checking in automated recognition of human activities for ambient-assisted living. *IEEE Trans. Hum. Mach. Syst.* **2013**, *43*, 509–521. [CrossRef]
39. Ye, J.; Coyle, L.; Dobson, S.; Nixon, P. Ontology-based models in pervasive computing systems. *Knowl. Eng. Rev.* **2007**, *22*, 315–347. [CrossRef]
40. Riboni, D.; Bettini, C. Context-aware activity recognition through a combination of ontological and statistical reasoning. In Proceedings of the Ubiquitous Intelligence Computing, Brisbane, Australia, 7–9 July 2009; pp. 39–53.
41. Azkune, G.; Almeida, A.; López-de Ipiña, D.; Chen, L. Extending knowledge-driven activity models through data-driven learning techniques. *Expert Syst. Appl.* **2015**, *42*, 3115–3128. [CrossRef]
42. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2010.
43. Augusto, J.C.; Nugent, C.D. The use of temporal reasoning and management of complex events in smart homes. In Proceedings of the 16th European Conference on Artificial Intelligence, Valencia, Spain, 22–27 August 2004; Volume 16, p. 778.
44. Dumas, M.; La Rosa, M.; Mendling, J.; Reijers, H.A. *Fundamentals of Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 1.
45. Gottfried, B.; Guesgen, H.; Hübner, S. Spatiotemporal Reasoning for Smart Homes. In *Designing Smart Homes*; Lecture Notes in Computer Science; Augusto, J., Nugent, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4008, pp. 16–34.
46. Aztiria, A.; Izaguirre, A.; Basagoiti, R.; Augusto, J.C.; Cook, D.J. Discovering Frequent Sets of Actions in Intelligent Environments. In *Intelligent Environments*; IOS Press: Barcelona, Spain, 2009; pp. 153–160.
47. Aztiria, A.; Izaguirre, A.; Basagoiti, R.; Augusto, J.C.; Cook, D. Automatic modeling of frequent user behaviours in intelligent environments. In Proceedings of the 2010 Sixth International Conference on Intelligent Environments (IE), Kuala Lumpur, Malaysia, 19–21 July 2010; IEEE: Washington, DC, USA, 2010, pp. 7–12.
48. William, C.; others. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 115–123.

49. Bettini, C.; Brdiczka, O.; Henricksen, K.; Indulska, J.; Nicklas, D.; Ranganathan, A.; Riboni, D. A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.* **2010**, *6*, 161–180. [CrossRef]
50. Sadri, F. Ambient Intelligence: A Survey. *ACM Comput. Surv.* **2011**, *43*, 36. [CrossRef]
51. Chen, L.; Hoey, J.; Nugent, C.D.; Cook, D.J.; Yu, Z. Sensor-based activity recognition. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2012**, *42*, 790–808. [CrossRef]
52. Acampora, G.; Cook, D.J.; Rashidi, P.; Vasilakos, A.V. A Survey on Ambient Intelligence in Healthcare. *Proc. IEEE* **2013**, *101*, 2470–2494. [CrossRef] [PubMed]
53. Augusto, J.C.; Callaghan, V.; Cook, D.; Kameas, A.; Satoh, I. Intelligent environments: A manifesto. *Hum. Cent. Comput. Inf. Sci.* **2013**, *3*, 12. [CrossRef]
54. Amiribesheli, M.; Benmansour, A.; Bouchachia, A. A review of smart homes in healthcare. *J. Ambient Intell. Hum. Comput.* **2015**, *6*, 495–517. [CrossRef]