information

MDPI

Article

User in the Loop: Adaptive Smart Homes Exploiting User Feedback—State of the Art and Future Directions

Abir B. Karami, Anthony Fleury *, Jacques Boonaert and Stéphane Lecoeuche

University of Lille, F-59000 Lille and Mines Douai, URIA, F-90508 Douai, France;
Abir.Karami@mines-douai.fr (A.B.K.); Jacques.Boonaert@mines-douai.fr (J.B.);
Stephane.Lecoeuche@mines-douai.fr (S.L.)
* Correspondence: Anthony.Fleury@mines-douai.fr; Tel.: +33-3-27-71-23-81

Academic Editors: Sugam Sharma and Anna Fensel
Received: 12 April 2016; Accepted: 3 June 2016; Published: 15 June 2016

Abstract: Due to the decrease of sensor and actuator prices and their ease of installation, smart homes and smart environments are more and more exploited in automation and health applications. In these applications, activity recognition has an important place. This article presents a general architecture that is responsible for adapting automation for the different users of the smart home while recognizing their activities. For that, semi-supervised learning algorithms and Markov-based models are used to determine the preferences of the user considering a combination of: (1) observations of the data that have been acquired since the start of the experiment and (2) feedback of the users on decisions that have been taken by the automation. We present preliminarily simulated experimental results regarding the determination of preferences for a user.

Keywords: smart homes; user-centered decision making; Markov decision process

1. Introduction

Research on smart homes and on ambient-assisted living has grown these last years. This is due mainly to the low price of sensors and actuators and their facility of installation. This brought some new applications, using data mining techniques, to improve the way of responding of automation systems.

An application appeared these last year with the evolution of the population in developed countries. The augmentation of quality of life and of the capacity of the medicine made people live longer. However, problems appear when the person starts losing autonomy. Lots of research nowadays concentrate on the use of ambient assisted living architectures to monitor elderly people at home. This is done using, for instance, activity recognition to evaluate the performances of Activities of Daily Living (ADL).

Added to this work, quality of life of the elderly people in their home can also be improved by helping them in their daily life, using automation systems. However, elderly people can quickly be bothered by the technology. To be usable, especially by this very demanding population, the automation has to be adapted to the user and to his/her way of living. Human feedback or signals of their level of satisfaction are important sources of information for adaptive and personalized systems.

Most existing studies in adaptive systems and smart environments concentrate on user independent approaches. In such applications, it is expected that the system takes into account the variety of users (with different ages, preferences and needs, *etc.*). Thus, we are interested in this work in a user dependent approach. For personalized adaptive systems in multi-user applications, it is difficult to manually predict the adaptive knowledge for all possible situations. Furthermore, in certain scenarios, it is practically impossible for the designer to define the needed knowledge (adaptive knowledge considering the different possible situations and the different users' preferences) that covers all possible user profiles and suits all possible situations. For this reason, we argue that such knowledge including users' preferences (represented in a reward function) can be learned through analyzing the history of interactions. The proposed approach works under the assumption that users can provide positive and negative feedback regarding the system actions. The user feedback can be collected from facial expressions, gestures, vocal expressions, *etc*.

To address the decision making process, we propose to implement a Markov Decision Process (MDP) in which the set of states of the MDP represents low level information such as sensor readings and high level information such as detected currently performed activities (determined by a classification algorithm) and the users' updated profiles. The MDP calculated policy will be responsible of choosing the best action knowing the current state. We argue in this work that approaches that detect the relevance of each situation attribute (environment and user profile attributes) for each possible action learn faster than those that does not detect such relevance. This helps in generalizing the reward function and allows the system to adapt to new situations (e.g., with new users), and more importantly helps in decreasing the complexity of convergence to an optimal adaptive policy.

The major contribution of this work is to create such a system. It will gather user inputs (feedback), sensor readings from the environment, and knowledge from previous interactions. The sensor readings can be used by both an activity recognition system to create higher level information and by an adaptive decision process updating its reward function. From all this information, we expect to build an enough representative knowledge that will allow the system to adapt to its users' preferences.

The organization of this article is as follows. First, Section 2 will be devoted to the state-of-the-art of smart homes, activity recognition and user feedback in smart environments. Then, in Section 3, the description of the adaptive and personalized smart home system will be presented. In Section 4, preliminary results concerning the determination of preferences for the users will be introduced. Section 5 will present the future work and a description of an on-going experiments in our living lab to reinforce the first results. Finally, Section 6 will conclude this article.

2. State-of-the-Art

2.1. Smart Homes

In the 1970s, the use of actuators in homes for automation started to appear. This idea has grown over the years and the technologies were more and more developed. These systems use their own bus or wireless connection for a full installation. They can also use advanced technologies such as energy harvesting for transmission efficiency and simplicity of use.

Later, in the 1990s, the idea of adding sensing technologies to these systems has been explored, to create what we now call "Smart Homes". The evolution is that the analysis of the sensor measurements are made to create a more adapted behavior of the automation. It is the concept of Ambient Assisted Living. The use of these sensing technologies can have different applications [1]. The first is for health, using the measurement to infer the well-being of the person; the second is the energy efficiency, with the use of the data to regulate the energy consumption of the house with the most possible accuracy; and the third is security of the materials and of the person inside. For these applications, a large set of sensors could be used and lots of projects include different sensors and the appropriate processing on their data. It includes now smart objects (in IoT context) [2]. An international selection of leading smart home projects is presented by [3], as well as the associated technologies of wearable/implantable monitoring systems. Another state-of-the-art article concentrated on the utilized techniques in smart homes [4]. The article presents projects using audio-based, video-based, audio-visual-based, sensor-based and multi-modal-based techniques for smart home activity detection methods.

A very rich analysis of the existing work in smart homes including a study of a classification criteria for more than 154 publications is presented in [5]. The classification yields four main clusters regarding four aspects; the financial aspect with a market and financial analysis, the service specification and service design aspect, the organizational aspect as in partnerships and commercialization, and finally the technological aspect as in design, middle-ware, architecture, smart technologies, application areas, *etc*.

Another article presents smart homes' projects regarding comfort, health-care and security [6]. They summarized, in an interesting table, the taxonomy of home and user monitoring equipment and devices in current and past projects. Another summary table lists currently used methods and algorithms.

In [7], an analysis of literature reviews on smart homes and their users through three themes is presented: (1) functional, instrumental and socio-technical view of the smart home; (2) users and prospective users and their abilities to interact and use the technologies in the home; (3) hardware, software, design and domestication challenges for realizing a smart home. Regarding the second point, the authors reveal, through their analysis, a notable absence of user-focused research: the characteristics of prospective smart home users; how they might interact with the smart home through its technologies; and how the conceptualizations of the home and its technology can affect the usability and acceptability.

An important application of Smart Home technologies, in addition to their use in health or safety applications, is the comfort improvement. Visher [8] defines the comfort with a pyramid and three different levels to measure/improve it. The first one, the basis of the triangle, is Physical Comfort. It is the basis for habitability. It is assured by responsible building design and operation, as well as by setting and meeting standards of health and safety. What is in line here is the temperature, level of noise, air quality, security of the construction, *etc*. The second one is the functional comfort that addresses the way the installation helps the user to live his/her life. Finally, the last level, at the top of the pyramid, is the psychological comfort. [9] presents it saying that it "*relates to human needs and lifestyles. This comfort is more related to the conditions of integration of technology with space for better compatibility with a user's everyday life; here, technology push is replaced by a user center approach.*" This last article revisits the works about smart homes with the view of this pyramid and with what it can bring as comfort to the person. The final goal of our work is to address the psychological comfort by improving the user experience in his/her own home.

Another trending topic that we could add to this state-of-the-art is assistive robotics to complete the smart home experience. According to [3], assistive robotics is now considered as a component of large smart home environments. Robots provide useful services and act as companions to ease the burden of social isolation. The authors discuss the two principal research directions in robotics: task-oriented robots and interaction-oriented robots. Several robotic projects are discussed in this review including robots to assist the elderly, the disabled, and children: navigation chair systems, cooperating arms, Nursebots, rehabilitation assistants, and entertainment robots. In [10], the authors present a review of the current works that are related to the use of these assistant robots for elderly people that have to cope with dementia.

In this paper, we concentrate on *adaptive smart homes* and the users' possibilities to interact with them. As we show later in this section, recent research exploits user feedback and interactions with the smart home to increase its intelligence. Our goal is to go further to improve user experience in multi-inhabitant smart homes.

2.2. Activity Recognition

Activity recognition is a topic of interest for automation and health applications in smart homes. It can be used for different applications. One of them is to adapt automation of the home to the circumstances of the command that is perpetrated. For instance, turning on the light can be done differently if it is to wake up the person in the morning or if it is to give the person some more light for cooking when the outside luminosity has decreased too much.

A relatively complete state-of-the-art of activity recognition can be found in [11]. It also lists the datasets that have been acquired to work on this topic and to create models of activity.

Activity recognition is generally done using supervised classification algorithms. However, two types of algorithms can be found to perform this task:

- Offline classification algorithms, such as K-Nearest-Neighbors (K-NN) [12], Artificial Neural Networks (ANNs) [13] , Decision Trees (DTs) [14] or Support Vector Machines (SVMs) [15]. These algorithms rely on the creation of "frames" of data of a chosen length that will try to find the closest example(s) in the database for a test frame. These algorithms are based only on statistical evaluation of the dispersion of the data in a given space. Using ontologies, [16] defines the context and the Activities of Daily Living for a further recognition with rules-based algorithms.
- Sequential algorithms, such as Hidden Markov Models (HMMs) [17], Conditional Random Fields (CRFs) [18] or also Markov Logic Networks (MLNs) [19]. Those methods add to the previously cited algorithms a notion of dependence between the different events of the frame or of the activity. This allows to identify spatio-temporal relationships between the data that are totally absent in the classical methods. HMMs have been, for a long time, a reference method for activity recognition.

For all these algorithms, the conception of a dataset that is annotated is mandatory to obtain the models of each activity and to be able to test the performances. In the results that are obtained, the models are relatively generic as they are not dependent of the person performing the activity. However, they could depend on the sensors that are present in the environment and their disposition. To perform the classification, one important step is to compute some relevant "features" (statistics on the signal that allow for differentiation of the activities and that are common between realizations of the same activity). Depending on the setup and on the set of sensors of the smart home, this step has to be adapted, and the features will differ.

Another problem of activity recognition is being able to perfectly separate the different activities. The problem is addressed in [20] in which the authors use a first step of clustering before classification of activities. Considering another approach, in [19], we decided to create another class (unknown class), in which we aim to put all the transitions between activities and also overlap the activities. This, as a consequence, will decrease the performance of classification as some samples of this new class will partly contain ones of each of the other classes.

Finally, another approach consists in using adaptive and online learning algorithms (such as an adapted version of SVM) to classify each firing of sensors. Considering the last firings with a weighting factor, we give less importance to events that occurred a long time ago. This approach has been tested in [21].

These works show that there is a huge dependency on the setup and on the type of used sensors. Rashidi *et al.* [22] propose using a mining algorithm called an Activity Adaptation Miner (AAM) along with guidance-based learning and observation-based learning. This proposition forms reinforcement learning to discover changes in previous automated activities and adapt to user advice.

2.3. User Feedback for Smarter Homes

An important aspect of smart home research concentrates on the users' well-being and level of comfort. Smart homes should be able to understand the needs of their occupants (users) and adapt/personalize the decisions accordingly. However, instead of manually defining and updating users' preferences, a smart system can use the feedback to update their preferences automatically and learn how to adapt to them.

Intelligent environments adapt their behavior to satisfy the users by learning patterns in their behavior. This implies that user's feedback should be collected either *implicitly* (normal operation of standard devices like turning off the lights) or *explicitly* through friendly interfaces [23,24]. Interacting with smart systems should be simple, self-explaining, intuitive or even natural without the need to educate or train users in a special way. The definition and development of intuitive user interfaces is an ongoing search [25], however not the subject of this paper.

Human feedback is used differently, by diverse approaches, for planning processes. Some approaches use human feedback as shaping signals to teach a system how to achieve a task. In such approaches, the source of the feedback is considered as an observer of the system who evaluates each of the system's actions [26,27] or the system's entire policy [28]. The TAMER (Training an Agent Manually via Evaluative Reinforcement) framework [26] proposes a method to shape a learning robot by giving positive and negative signals (as for a domestic dog). This method helps in training the robot to execute a task. These approaches are not proposed to learn personal preferences nor to be used in multi-user environments like smart home environments.

The experience feedback loop approach [29,30] uses a case-based reasoning system to choose the best matching case to reuse. Afterwards, the user gives *a priori* feedback about this choice. If the case does not work, the user can modify the case (or adds a new one) or gives negative *a posteriori* feedback. A validity monitoring component collects the feedback and automatically calculates the validity function which is used for enhancing search result presentations and for automatically triggering maintenance for cases with insufficient validity. This approach, however, requires a complete supervision by the user to give the *a priori* and *a posteriori* feedback and adjust the rules when needed. Such supervision is highly unrecommended in smart home scenarios.

Another approach is proposed for autonomous learning of a user's preference of music and light services in smart home applications [31]. The proposed approach includes a web based platform available through a personal computer or a personal digital assistant that has the ability to get the user's reward explicitly or to calculate it implicitly. The adaptive system controls automatically the music and the lighting in the smart home by adjusting six different decisions (genre, mood, volume, pattern, color and brightness) using 41 different actions. The state of the environment is represented with the following attributes (current time, user activity and user location). Time is represented by 288 time slots of 5 min a day. A function approximation technique (tile coding) is used to group neighbor states in time and propagates evidence from a state to another. An unsupervised reinforcement learning method (Temporal Differential) is used to learn the user's preference dynamically from feedback and perform the optimal actions.

An adapting mobile interaction obtrusiveness by exploiting user feedback is presented in [32]. They present a method for adapting interaction obtrusiveness automatically based on two explicit types of feedback (reward or punishment). Instead of asking the user to redefine his preferences about interaction obtrusiveness configurations, they learn them by means of the received feedback, using a Q learning algorithm, in a way that maximizes user's satisfaction in long-term use. The paper does not represent a direct application for smart homes but for connected devices used in smart home environments.

The CASAS (Center for Advanced Studied in Adaptive Systems) project including the CASA-U (the CASAS User interface) [22,33] introduces an adaptive smart home system that utilizes machine learning techniques to discover patterns in resident's daily activities and to dynamically adapt to the user's explicit or implicit wishes in daily routine activities. CASAS can also adapt to the changes in discovered patterns based on the resident implicit and explicit feedback and can automatically update its model to reflect the changes.

Our on-going research project is very close to those presented lastly [22,31–33]. However, we are interested in smart homes that are able to adapt to different users by learning the preference of each of them using their individual implicit and explicit feedback and those of others. We propose in this paper a global architecture that integrates users' profiles to the learning and decision making processes. In this work, we will define an attribute as an information representing a part of the state of the environment (for instance level of brightness, temperature, *etc.*) or the user profile (age, gender, habits of living, *etc.*). Some of the user profile attributes will be learned automatically using the activity recognition system. For example, an attribute representing whether the person has a habit of practicing sport frequently can be learned/updated from the readings.

Our approach is based on the automatic detection of the relevance between the learned preference (the user *x* prefers watching news media while practicing inside sports) and the attributes representing the environment (rainy day) and the user profile (practices sports frequently). Such relevance is an important key element for generalizing the learned adaptive behavior to new situations and unknown users and for decreasing the complexity of convergence to an optimal adaptive policy (decreases the size of needed dataset to reach an acceptable adaptive automation). Our work will include personalized automation on the use of light, on the control of the temperature of the home, on the media consultation (TV, radio, music, *etc.*). Our goal is to be able to use all previous interaction in addition to current measurement in the home to adapt the controls of all of the previously mentioned actuators depending on the profile of the current user (and also to be able to slightly update the preferences if the decision does not fit with the user expectations).

3. Adaptive and Personalized Smart Home Behavior

3.1. Global Architecture

Figure 1 describes the different components (processes and databases) of the proposed architecture and shows the data flow from acquisition to decision making. The architecture contains five databases and three processes. The first database (called "raw data"), taking information from the smarthome directly, contains all the sensors' readings and all the profiles of the different users (the current version). A user profile can be partially completed directly by the user and then, while processing the available data, can be completed/updated automatically.

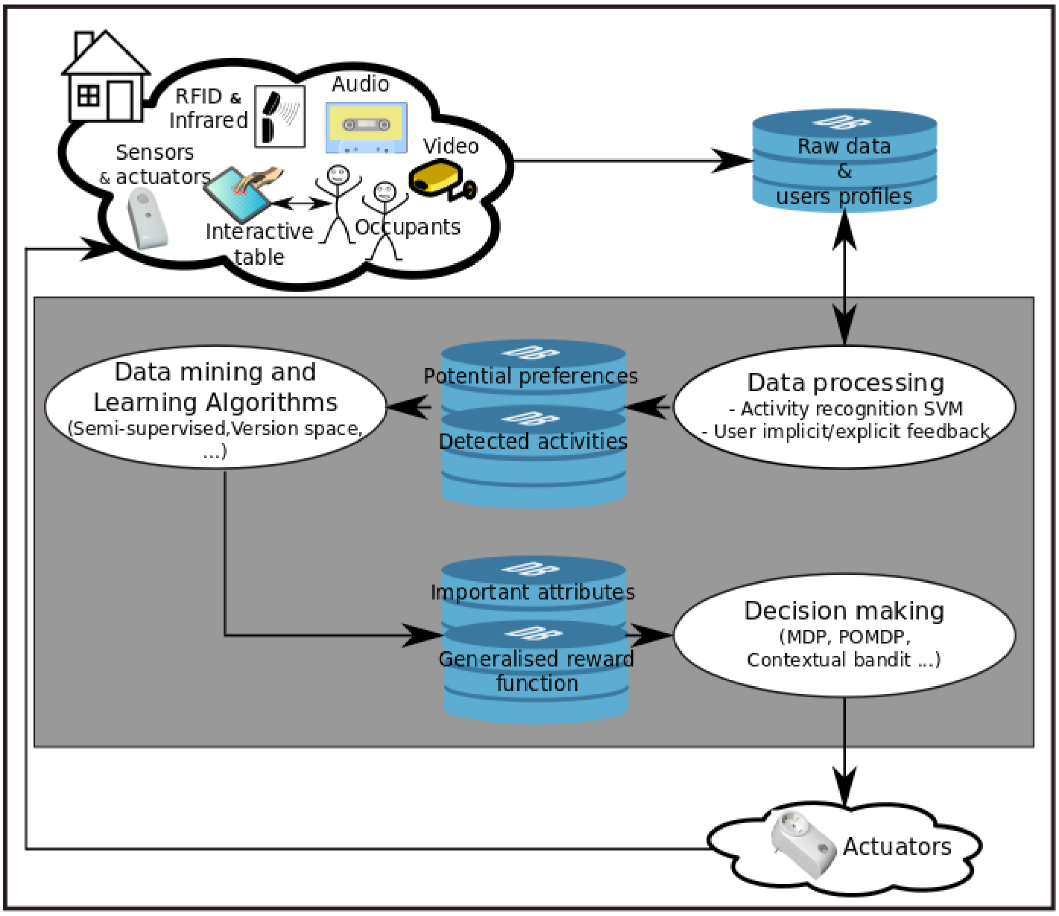


Figure 1. Description of the global architecture of the system, from raw value acquisition to decision making.

The "data processing" process is responsible for three missions:

1. activity recognition,

2. detecting implicit feedback, and exporting potential preferences through analyzing each received explicit feedback and each detected implicit feedback, and
3. updating users profile using the detected new habits.

The process of activity recognition analyses the sensors readings from the database to determine an activity that is occurring in the home. The raw database is analyzed also to detect the potential preferences represented by a set of (situation, action, feedback) triples. A potential preference is registered when:

1. an automated action by the smart home followed by a user explicit feedback is encountered, and
2. when implicit feedback is encountered (a change of an actuator value by the user him/herself).

Those mentioned processes will fill two databases called "potential preferences" and "detected activities". These databases will be the entry for the data mining and learning processes that are responsible for updating the learned reward function of the Markov Decision Process. During this learning process, the learned reward function is generalized to cover unknown situations and new users by determining the key attributes for each possible action. The detected important attributes and reward functions are saved in two eponymous databases. The last process is responsible for deciding and controlling the actuators to change the environment state using the action chosen by the MDP policy. The components and their functionality are described in detail in the following part of this section.

Potential preferences are registered in a database and used to learn or update the generalized reward function (generalized adaptive rules). We use, in this work, methods of generalization that are based on detecting important attributes for each decision (see Section 3.6). The decision making process uses the generalized reward function to plan optimal decisions that are adaptive to the current situation. We refer to the current situation as both the current state of the smart home and the profile of the user that the smart home is observing at the current time.

For example, let us consider the action of turning on the light. Such an action depends on the level of brightness in the room, which is an important attribute regarding this action. The process of learning will detect this attribute as important with time. However, a decrease of brightness in the room could be measured when the person is currently sleeping or when he/she is reading a book. In the first case, it is not appropriate to turn on the light, and, in the second, it is preferred. In both cases, the user preferences learned by analyzing his/her feedback include that, with a very low brightness, it is preferred to turn on the light. However, through the activity recognition system, this architecture could prevent an undesired behavior. In this work, the activity recognition output complements the other entries of the decision process.

3.2. Description of the Installation of Sensors

Figure 2 shows a general view of our Living Lab installation. Sensors and actuators in this double room living lab are:

- Sensors for all the possible electrical plugs of both rooms,
- Light, humidity and temperature sensors at different positions of the lab, and others for the outside,
- Sensors to measure the position of the shutter, the opening/closing of the doors and of the windows,
- Infra-red sensors, positioned at strategic locations in the rooms, to know where the person is (for activity recognition purpose, for instance),
- Actuators to command the different heaters of the rooms,
- Actuators to command the light (in four sets of lights distributed in the room),
- Actuators to turn on/off the different electrical outlets/plugs,
- Actuators to change the positions of the windows shutters,

- A sensor on the door that will, with an access card, recognize the person entering the room so that the system can load his/her profile (if already filled).

For experimental purposes, we installed some commodities:

- Table and chairs to sit comfortably,
- A large smart TV to have access to different media such as music, TV, radio, YouTube channels, Internet, *etc.*,
- A baby foot table to play (several persons can be present in the living lab at the same time),
- A smart table using Microsoft Windows to allow people to use internet, read, *etc.* but also to give a feedback regarding smart home automation (see following sections).

Pictures of these installations are given by Figure 3.

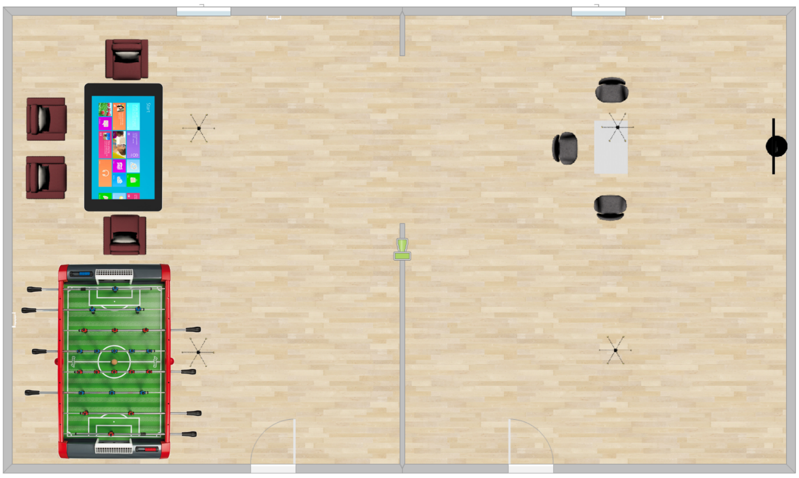


Figure 2. Ambient environment of Mines Douai.

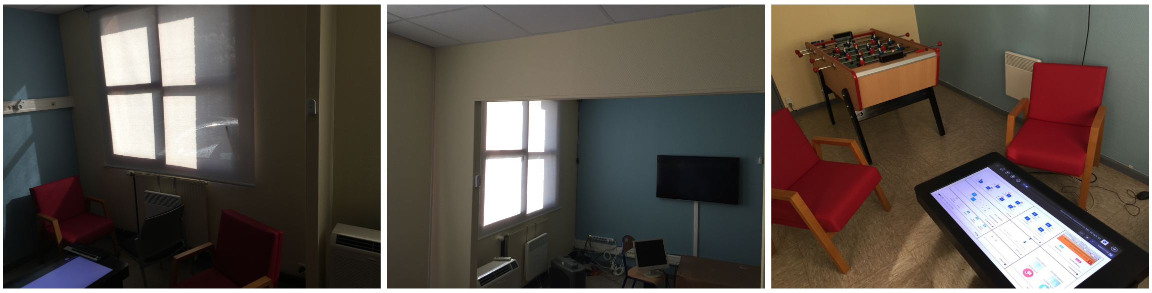


Figure 3. Installation in Mines Douai Living Lab.

3.3. Data Processing and Acquisition

The data of the different sensors and the results of the actuators' actions are acquired wirelessly (all the sensors are connected in a Zigbee [34] network). A Java-based software (Cleobee, V5.10, CLEODE, Lannion, France) [35] is responsible for the acquisition of the sensors and the integration of all the data into a MySQL database that is kept on a computer in a technical room near the living lab. The acquisitions are centralized on only one computer by this software. The software is responsible for receiving, ordering and time-stamping these readings; therefore, no synchronization is needed.

We construct, from these readings of sensors, what we call a Raw Data Element (RDE), which is defined as follows.

Definition 1. A non-exhaustive description of an RDE can be represented in the tuple $\langle id, type, value, timestamp \rangle$, where:

- id is the identifier of the RDE in the database,
- type is the description of the type of the RDE,
- value describes the value of the RDE,
- timestamp is the timestamp describing the time that the changing of the RDE value took place.

Example 1. An RDE of the type "level of brightness" can have a value $\in [0, 1000]$, an RDE of the type "feedback" can have a value $\in [-1, 1]$ and an RDE of the type "Kitchen light switch" can have a value $\in \{on, off\}$.

The interactive table (Figure 4) has a web interface that can allow the person to perform the following actions:

- Check the state of the different sensors of the room and have information on the environment,
- Perform actions on the room such as turn on/off lights, power outlets, *etc*.
- Give an opinion on an action that just occurred in the room. For instance, if the smart home turns on the lights after sensing a decrease in the ambient luminosity and the user does not agree for some reason, he/she can notify it on the screen (thumbs up, thumbs down buttons).

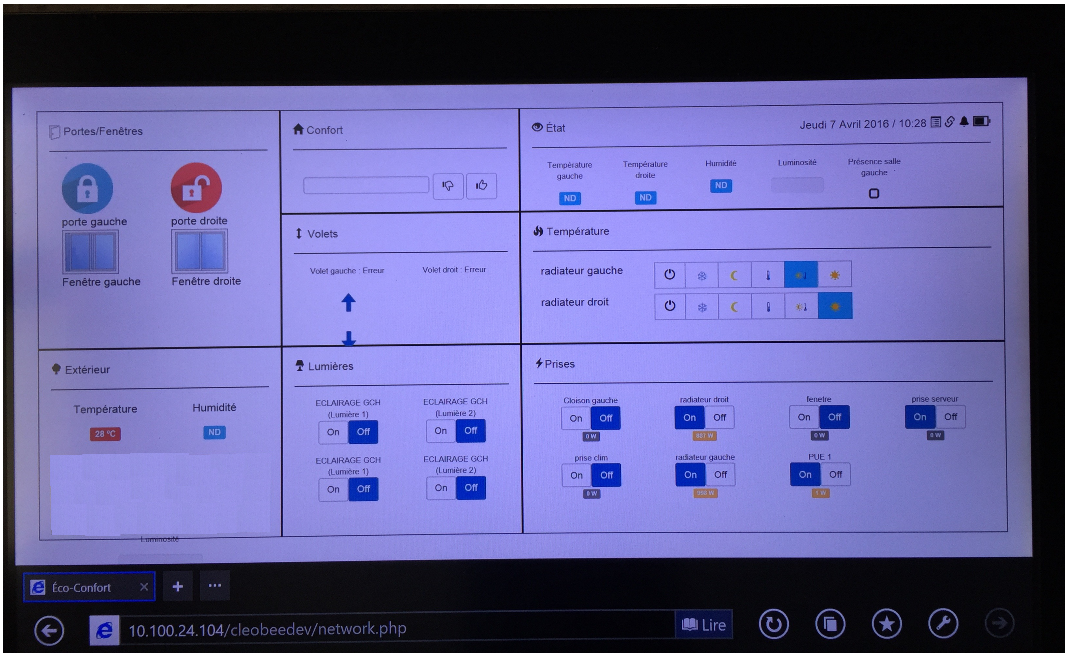


Figure 4. Ambient environment of Mines Douai—a web application shown through an interactive table.

All actions performed on the table are registered as RDEs. A change in the value of any actuator through the interactive table web application or manually is considered as explicit feedback and an opinion feedback through the satisfaction buttons (thumbs up and down) is considered as implicit feedback (as for vocal "yes" and "no" feedback).

Contrary to implicit feedback, explicit feedback (e.g., turning on the lights few seconds after it was turned off by the smart home) is easier to detect because the feedback can be easily connected to the action that causes it (the same actuator). When explicit feedback is received, a negative reward is associated to the smart home action with the current state. However, when implicit feedback occurs, the smart home action that causes it is not trivial to find. It is possible that different automated actions occurred a few seconds before the reception of the implicit feedback. For this reason, in our proposal, we work under the following assumption:

Assumption 1. A user implicit feedback o_u is received after the smart home action and is quantified using the function $\mathcal{V}$. $\mathcal{V}$ is a predefined value function that assigns a positive, negative or null weight for each feedback $\mathcal{V} : \mathcal{O}_u \to [-1, 1]$ (for example, $\mathcal{V}(\text{thumbs\_up}) = \mathcal{V}(\text{yes}) = +1$ and $\mathcal{V}(\text{thumbs\_down}) = \mathcal{V}(\text{no}) = -1$). A lack of feedback for an action can be quantified as positive, negative or null value depending on the scenario or learned user habits. Considering the problem of the interval between the smart home action and the user reaction, it is possible to define a function $\mathcal{V}(o^i) = h(\mathcal{V}(o^1), \ldots, \mathcal{V}(o^i))$ using a heuristic (e.g., propagation) [26,36].

3.4. Activity Recognition

In a previous work [19], we obtained some results on two different databases [37,38] with different methods. The results are summarized in Table 1.

Table 1. Overall accuracy (%) results on the two datasets with and without the Unknown class. Table 3 of [19]. the (diff.) column represent the different between both conditions (with and without unknown class).

Model	SH Dataset			HIS Corpus		
	Without Unknown	With Unknown	diff.	Without Unknown	With Unknown	diff.
SVM	75.00	71.90	3.10	74.86	64.90	9.96
Random Forest	82.96	80.14	2.82	70.72	62.32	8.40
MLN naive	79.20	76.73	2.47	75.45	66.81	8.64
HMM	74.76	72.45	2.31	77.26	67.11	10.15
CRF	85.43	83.57	1.86	75.85	69.29	6.56
MLN	82.22	78.11	4.11	75.95	65.82	10.13

Considering the available information and the different sensors that we have, we are closer to the HIS (French Health Smarthome) dataset than the SH (Sweet-Home) dataset presented in this publication. As a consequence, the right part of Table 1 shows us that the results between all the methods are relatively close to each other. As we have, since this work, improved our results with SVM, and as the implementation and the learning are largely faster than the other, we will focus on this method for our activity recognition part.

The computed features will also be of the same kind as the ones used in the previously mentioned publications, that is to say statistical variations of the values of the different sensors (mean, number of firings, *etc.*). A study of the importance of each feature in these conditions will also be performed in future work.

3.5. Adaptive Decision Making

Most approaches for adaptive systems use models based on probability theory and statistics, like Markov decision models. In the literature, a hierarchical activity model (HAM) which is a hybrid model of Markov decision processes combined with decision trees was presented for the CASAS smart home [22]. HAM utilizes contextual information such as temporal relations, start time distributions, duration distributions and startup triggers in order to predict and schedule automated activities accordingly.

In the architecture shown in Figure 1, the decision making component can use any decision process that allows the integration and the use of the learned rewards (adaptation rules). For simplicity and generality, a classic Markov Decision Process (MDP) can be used. However, other decision processes might be more appropriate depending on the framework properties, like, for example, Partially Observable Markov Decision Processes (POMDPs) [39] for partially observable environments or contextual multi-armed bandits [40] for one-step decision processes.

Formally, an MDP is represented by a tuple $\langle S, A, T, R \rangle$ where: S is a finite set of states; A is a finite set of agent's actions; T is a state transition function with $T(s, a, s') = Pr(s_t = s' | s_{t-1} =$

$s, a_{t-1} = a)$ representing the probability of transitioning from state s to state s' after doing action a, and $\sum_{s' \in S} T(s, a, s') = 1 \forall (s, a)$; and R is a reward function mapping $S \times A \times S$ to a real number that represents the agent's immediate reward for making action a while being in state s and ending in state s'. The objective of the agent is to calculate a policy $\pi : S \rightarrow A$, which assigns, for each possible state, an optimal action that maximizes the long-term expected reward $E\left[\sum_0^\infty \gamma^t r_t\right]$, where γ is a discount factor and r_t is the reward at time t. There are several algorithms to solve an MDP—classically Value Iteration [41] and Policy Iteration [42]. The complexity of such algorithms is $\mathcal{O}(|S^2||A|)$.

We will detail the MDP model through the following example:

Example 2. The MDP state includes all needed pieces of information that are connected to the decision process. This information is represented as a set of attributes (e.g., temperature sensor $\in [-50, 50]$, or user gender $\in \{male, female\}$) and their values. A state S can represent raw data values (e.g., sensor values, user profile) or a high level state after data treatment (e.g., present users and their current activities). The set of actions A represents all possible actions that can be controlled automatically by the smart home (e.g., turn on the lights or close the shutters). The transition function represents all possible changes in the smart home whether made by its last action $a \in A$ or by the current users. Finally, the reward function represents the expected received feedback for doing an action a in a current situation represented in the current state s (e.g., a state representing a low level of brightness and user is reading might probably lead to a positive feedback if the smart home turns on the lights). After calculating the MDP policy, the latter should be able to guide the system through actions that maximize the long-term expected rewards/feedback, satisfying, by this, the users' learned preferences.

We note that the reward function has the same representation of a potential preference; however, through the learning process, the attributes' values can be generalized or described as constraints (logical expressions) over their possible values in the reward function (e.g., user gender = * meaning that whatever the value of this attribute the reward function can be applied).

The aim of using an MDP to plan the smart home decisions is to produce the desired adaptive behavior according to the situation (thus personalized to the user). However, a correct MDP reward function that represents the real expected feedback for each (state, action) couple is needed to generate such behavior. This leads us to the problem of learning the reward function by analyzing the potential preferences including users' feedback, a subject that we will discuss in the following.

3.6. Learning from User Feedback

For this problem, our approach is based on the analysis of the potential preferences which represents the users' feedback towards the smart home actions in a defined situation. Each potential preference is analyzed by the system to update the reward function. Our approach also generalizes the rewards by learning the dependence between the smart home behavior and certain situation attributes (e.g., changing the shutter position depends on the level of brightness of the room but does not necessarily depend on the gender of the current user). This helps the system to better adapt to new users and to new situations.

The learning process allows the system to learn a generalized reward function by analyzing the potential preferences (user feedback over the system actions). We present briefly the Generalized Version Space (GVS) algorithm (detailed in [43]).

The used mechanism in this algorithm is inspired from the version space generalizing and specializing techniques. The version space [44] is a machine learning approach used for binary classification. Its major drawback is its inability to deal with noise, which means that any detected contradiction can cause the version space to fail in the learning process.

The GVS algorithm learns from an input set of potential preferences. The output of the algorithm is the modified/generalized reward function. The algorithm is based on the detection of important attributes for each decision and uses this fact to generalize the learned reward function and therefore permit the system to converge with a much lower number of interactions (potential preferences). An attribute $at \in AT$ is considered important for an action a if the feedback value towards a depends on the value of at. The generalization of the learned rewards allows the system to adapt to unknown situations (new users or un-encountered states). The architecture backs up all potential preferences in the associated database, so there is no loss of information because of the generalization. The potential preferences database is continuously used in the process of detecting important attributes. The detection of important attributes is mainly triggered by the detection of a contradiction between a newly received potential preference and the generalized rewards.

The main drawback of such algorithms is the fact that they are risky until they converge to a correct generalization. The gain in the number of needed interactions and the risk in performance before converging are evaluated in our simulated experiments in Section 4.

4. Preliminary Results

We present in this section some performance and convergence analysis using a the GVS algorithm compared with a simple memorizing method without generalization (where the generalized learned rewards are equivalent to the potential preferences). Those results are based on simulated potential preferences (raw data and explicit feedback). In the following, we describe the procedure that we followed and we analyze the results.

4.1. Parameters and Simulation of Situations

In the experiments, we fixed the number of attributes representing the situation and user profile to eight attributes, and the number of actions to five. The number of possible values for each attribute was defined randomly between two and five values.

4.2. Simulation of Potential Preferences

A potential preference consists of a current state, an action and feedback. To simulate potential preferences, we simulated current states by giving a random value for each of the eight attributes, respecting the number of values of each of them defined in the parameters. To simulate the user feedback, we used some predefined rules of preference. The predefined rules were generated randomly based on a random number of important attributes for each action (between one and three). The selected important attributes were later compared with the detected important attributes by the algorithms.

Each simulated potential preference was based on a randomly chosen simulated current state. The MDP policy is called to choose the best system action. Then, the predefined rules of preference are used to simulate the user feedback over the MDP action. A predefined rule concerning the action a is applicable in a situation if the important attribute values of the rule have the same values in the randomly generated situation.

To balance between exploitation and the exploration during action selection, we followed the epsilon greedy method with $\epsilon = 10$. This means that with a 90% chance, the algorithms chose the best MDP action knowing the current reward function; however, with a 10% chance, a random action is chosen. Such behavior helps in exploring eventually a global minima instead of exploiting in a local minima [45].

4.3. Procedure

The procedure of evaluation is a loop of the following:

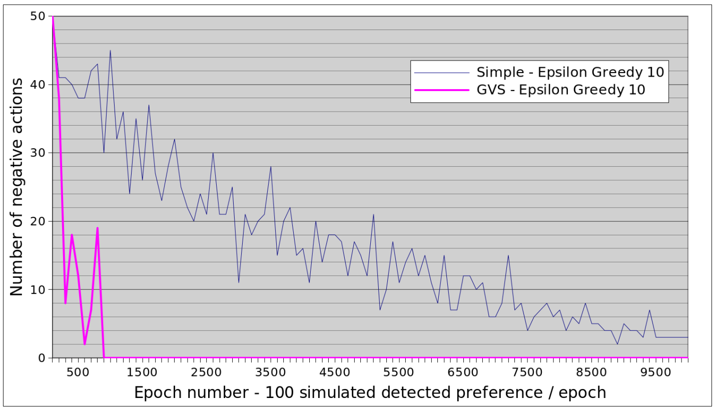
1. Re/Calculate the MDP policy.
2. Generate n potential preferences.
3. Evaluate the actions in the n potential preferences by counting the number of actions followed by a negative feedback called *negative actions*.

4. Learn/Generalize and update the MDP reward function using the n new potential preferences.
5. Repeat from step 1 until reaching max number of traces.

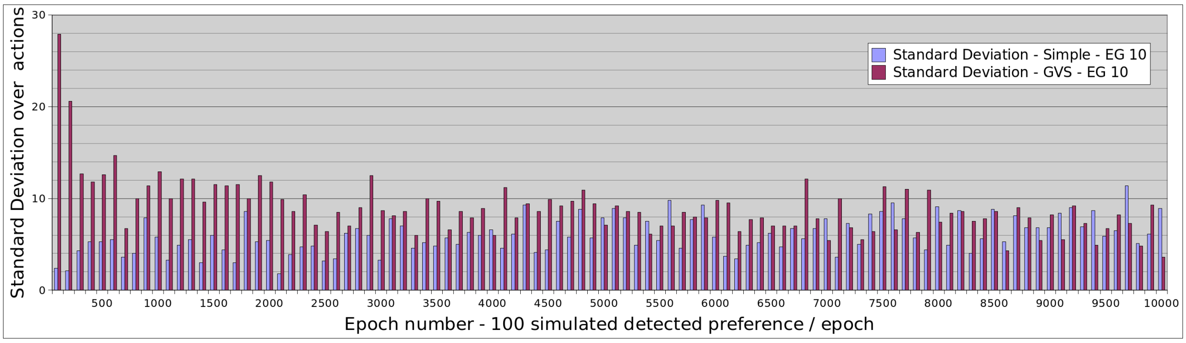
In the presented experiments, we set $n = 100$ and $max = 10{,}000$, which results in $10{,}000/100 = 100$ epochs of the previously mentioned five-step loop.

4.4. Convergence Results

We present a convergence analysis of the GVS algorithm compared with a simple one with no generalization algorithm. First, we analyze some results of simulated data using an epoch greedy mechanism with $\epsilon = 10$.



(**a**)



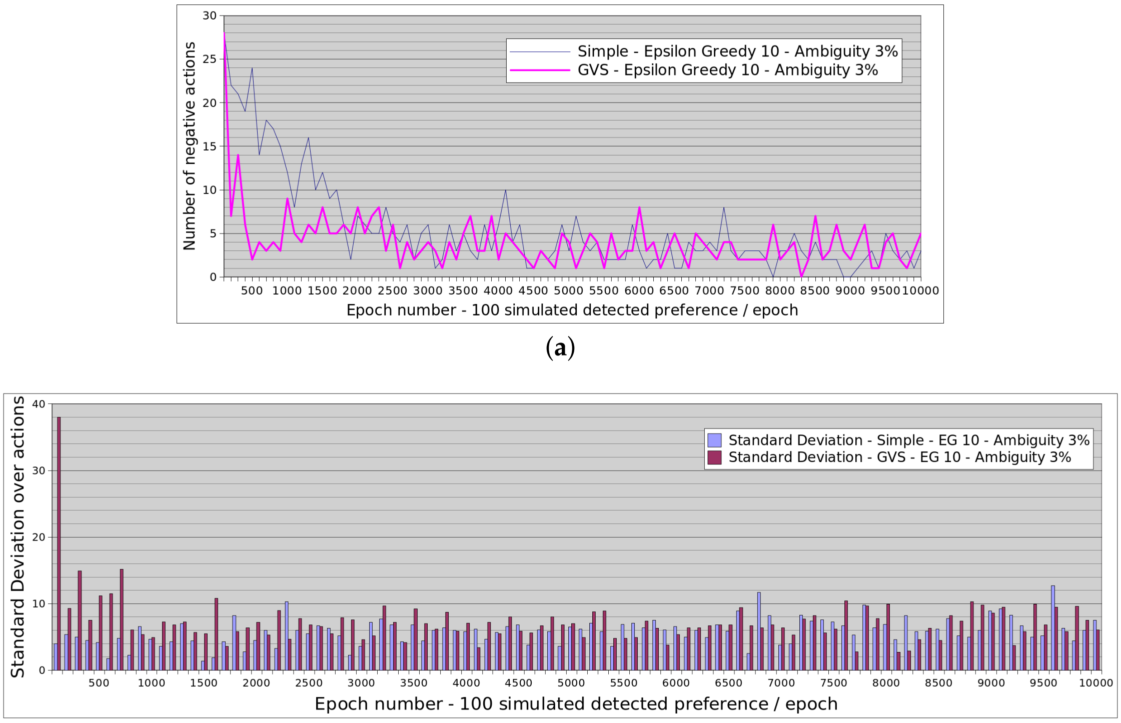
(**b**)

Figure 5. Simulated results comparing the GVS algorithm with a simple method using an epsilon greedy mechanism with $\epsilon = 10$ in the action selection process. (**a**) the number of negative actions in each epoch; (**b**) the standard deviation between the number of times each action was used during an epoch.

The upper schema of Figure 5 shows that the algorithm with generalization converges optimally after nine epochs (900 simulated potential preferences). However, the simple algorithm does not converge before 1000 epochs. It is important to mention that the randomly chosen parameters of this experiment led to a state space of 1944 possible states. The figure shows the number of negative actions (actions chosen by the algorithm and leads to a negative feedback given by the predefined rules of preferences). The number of negative actions is counted after generating the n potential preferences (step 3 in the procedure) where the learned rewards from the previous loop are used to generate the actions. The second graph of Figure 5 shows the standard deviation for the number of times each of the five actions were used. The value of this standard deviation demonstrates that different actions can be selected by the algorithms (and not always the same action). In the first few epochs, the algorithm with generalization generates a high standard deviation because it selects the same "with low risk" action most of the time. However, through learning and exploring (epoch greedy), the standard deviation decreases because other actions were found to be more advantageous.

Regarding the detection of the important attributes, the GVS was able to detect 100% of the important attributes. The random procedure to generate the experiments attributes chose two important attributes for the first and second actions, three important attributes for the third and fourth actions, and one important attribute for the fifth action.

Results of a second simulated experiment are shown in Figure 6. In this experiment, we test the capacity of the algorithm to deal with ambiguity in user feedback. During the simulation of potential preferences, we reversed the user feedback (negative if positive, positive if negative) with a probability of 3%. Results shows that the GVS algorithm converges faster than the simple one; however, as expected, it does not reach an optimal convergence (the predefined rules of preferences can not be regenerated through experience because of the ambiguity/noise in user feedback). The standard deviation analysis shows that, even with ambiguity in user feedback, the algorithm continues to choose interesting actions instead of repeating non-risky ones.



(**a**)

(**b**)

Figure 6. Simulated results comparing the GVS algorithm with a simple method using an epsilon greedy mechanism with $\epsilon = 10$ in the action selection process and with an ambiguity of 3% in users feedback. (**a**) the number of negative actions in each epoch; (**b**) the standard deviation between the number of times each action was used during an epoch.

These results prove the interest of detecting important attributes in the interaction situation for a better and faster adaptation. In previous work, a complexity and convergence analysis is presented for the GVS algorithm, in addition to a real experiment that proved the applicability of this algorithm for an adaptive and personalized behavior of a companion robot [43].

5. Future Orientations

5.1. Learning and Adaptation

In future work, we would like to automatically update/complete users' profiles by analyzing the interaction experiences and activity recognition component of our architecture. Such recognition permits the system to detect a change in users' habits (e.g., a user starts watching more TV). The update can be made automatically to the user profile (with/without his approval) and will permit an adaptation of the smart home decision to any change in users' habits. Such detection might also trigger other actions (e.g., report change of habit or anomaly to the caregiver in elderly care smart homes).

We have proven in our experiments that the detection of the important attributes helps the learning process to converge faster for personalized decision making in multi-user environments. Other approaches based on supervised and semi-supervised classification can also be applied. However, it is interesting to test their capacities with noisy information (ambiguity in the users feedback).

Current state-of-the-art approaches using user feedback lacks consideration for personalized behavior considering users' profile. The Cobot chat system proposed in [46] concentrates on adaptive behavior in multi-user environments using reinforcement learning. However, as the authors argue in their paper, the problem of learning from multi-users (not experts) feedback has certain properties, mainly that users have different characters, and, depending on their characters and the application itself, they tend to respond in rewarding or only penalizing feedback.

We are currently concentrating on studying the effectiveness of our proposed generalized algorithm in different framework structures (sequential long horizon planning and one-step decisions, penalizing or rewarding user characters, *etc*.) and comparing with the effectiveness of other methods as contextual bandit algorithms and decision tree based algorithms in each framework property.

5.2. Experimental Design

We are currently running preliminary experiments in the smart home with volunteer users. The objective of these experiments is to test the capacity of our proposed architecture to use users feedback and learn their preferences.

We focus in a first experiment on testing the wellbeing and the satisfaction of users regarding the level of brightness and the temperature of the room, the shutter positions, and proposed media (TV shows and radio) by using the list of sensors mentioned in Section 3.2 in addition to the identified user profile. In the experiments, we use explicit and implicit feedback through voice recognition, the web application on the interactive table and the detected direct actions of the user (e.g., user turning on or off the lights).

A first phase in the experiment will be essentially used to generate the raw data, in which the smart home will be mostly passive and will observe the explicit actions of the users. It will also be the moment at which we will be able to create the models of activities adapted to the environment that we have.

In a second phase, the potential preferences from the raw data collected in the first phase will be used to learn a generalized reward function (using the GVS algorithm), and the smart home will be more active while observing the users' implicit feedback and explicit actions to update the learned reward function. Users of the second phase will be asked to fill out a satisfaction questionnaire based on a Likert scale that analyzes their level of satisfaction towards the automatic behavior of the smart home and their opinion towards the way they interact with the smart home.

6. Conclusions

We presented in this paper a general description of existing research studies regarding smart homes in general and especially those that exploit user feedback to better adapt their behavior. We focus our interest on smart homes that are able to personalize their behavior (ambient intelligence) to the current users in multi-user environments. Such problems incite representing users by their profiles.

For the moment, our propositions can adapt to a user profile but does not handle several profiles at the same time.

We proposed a general architecture that describes the flow of information from data acquisition to decision making. Our approach is based on the fact that users' feedback is a rich source of information to learn to better adapt in the future. We presented our approaches for activity recognition (based on SVM) and learning adaptive ambient behavior based on semi-supervised learning algorithms that uses user feedback to learn a generalized reward function.

We presented some preliminary results based on simulated data. These promising results prove the importance of detecting important attributes for each action. We are currently running more experiments in our the Douai living lab to validate our approach and architecture in a real environment.

Acknowledgments: This work is funded by the French Ministry of Industry through Mines Douai.

Author Contributions: Abir B. Karami, Anthony Fleury, Jacques Boonaert and Stéphane Lecoeuche contributed equally to this work. All authors have read and approved the final manuscript.

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

References

1. GhaffarianHoseini, A.; GhaffarianHoseini, A.; Tookey, J.; Omrany, H.; Fleury, A.; Naismith, N.; Ghaffarianhoseini, M. The essence of smart homes: Application of intelligent technologies towards smarter urban future. In *Handbook of Research on Creative Technologies for Multidisciplinary Applications*; IGI Global: Hershey, PA, USA, 2016; Chapter 14, pp. 334–376.
2. Zhu, N.; Diethe, T.; Camplani, M.; Tao, L.; Burrows, A.; Twomey, N.; Kaleshi, D.; Mirmehdi, M.; Flach, P.; Craddock, I. Bridging e-Health and the Internet of Things: The SPHERE project. *IEEE Intell. Syst.* **2015**, *30*, 39–46.
3. Chan, M.; Estève, D.; Escriba, C.; Campo, E. A review of smart homes—Present state and future challenges. *Comput. Methods Progr. Biomed.* **2008**, *91*, 55–81.
4. De Silva, L.C.; Morikawa, C.; Petra, I.M. State of the art of smart homes. *Eng. Appl. Artif. Intell.* **2012**, *25*, 1313–1321.
5. Solaimani, S.; Keijzer-Broers, W.; Bouwman, H. What we do-and don't-know about the Smart Home: An analysis of the Smart Home literature. *Indoor Built Environ.* **2013**, doi:10.1177/1420326X13516350 .
6. Alam, M.R.; Reaz, M.B.I.; Ali, M.A.M. A review of smart homes—Past, present, and future. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2012**, *42*, 1190–1203.
7. Wilson, C.; Hargreaves, T.; Hauxwell-Baldwin, R. Smart homes and their users: A systematic analysis and key challenges. *Pers. Ubiquitous Comput.* **2014**, *19*, 463–476.
8. Vischer, J.C. The concept of environmental comfort in workplace performance. *Ambient. Constr.* **2007**, *7*, 21–34.
9. Allameh, E.; Heidari Jozam, M.; Vries, B.D.; Timmermans, H.; Beetz, J. Smart Home as a smart real estate: A state-of-the-art review. In Proceedings of the 18th International Conference of European Real Estate Society, ERES 2011, Eindhoven, The Netherlands, 16-18 June 2011.
10. Novitzky, P.; Smeaton, A.F.; Chen, C.; Irving, K.; Jacquemard, T.; O'Brolcháin, F.; O'Mathúna, D.; Gordijn, B. A review of contemporary work on the ethics of ambient assisted living technologies for people with Dementia. *Sci. Eng. Ethics* **2014**, *21*, 707–765.
11. Cook, D.J.; Krishnan, N.C. *Activity Learning: Discovering, Recognizing, and Predicting Human Behavior from Sensor Data*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
12. Fahad, L.G.; Ali, A.; Rajarajan, M. Learning models for activity recognition in smart homes. In *Information Science and Applications*; Kim, J.K., Ed.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 819–826.
13. Bourobou, S.T.M.; Yoo, Y. User activity recognition in smart homes using pattern clustering applied to temporal ANN algorithm. *Sensors* **2015**, *15*, 11953–11971.
14. Mckeever, S.; Ye, J.; Coyle, L.; Bleakley, C.; Dobson, S. Activity recognition using temporal evidence theory. *J. Ambient Intell. Smart Environ.* **2010**, *2*, 253–269.
15. 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.
16. 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.

17. Duong, T.; Phung, D.; Bui, H.; Venkatesh, S. Efficient duration and hierarchical modeling for human activity recognition. *Artif. Intell.* **2009**, *173*, 830–856.
18. Nazerfard, E.; Das, B.; Holder, L.B.; Cook, D.J. Conditional random fields for activity recognition in smart environments. In Proceedings of the 1st ACM International Health Informatics Symposium, IHI '10, New York, NY, USA, 11–12 November 2010; ACM: New York, NY, USA, 2010; pp. 282–286.
19. Chahuara, P.; Fleury, A.; Portet, F.; Vacher, M. On-line human activity recognition from audio and home automation sensors: Comparison of sequential and non-sequential models in realistic Smart Homes. *J. Ambient Intell. Smart Environ.* **2016**, in press.
20. Hoque, E.; Stankovic, J. AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities. In Proceedings of the 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), San Diego, CA, USA, 21–24 May 2012.
21. Yala, N.; Fergani, B.; Fleury, A. Feature extraction and incremental learning to improve activity recognition on streaming data. In Proceedings of the 2015 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), Douai, France, 1–3 December 2015.
22. Rashidi, P. Smart Home Adaptation Based on Explicit and Implicit User Feedback. Master's Thesis, Washington State University, Pullman, WA, USA, 2007.
23. Aztiria, A.; Izaguirre, A.; Basagoiti, R.; Augusto, J.C. *Learning about Preferences and Common Behaviours of the User in an Intelligent Environment*; Book Manufacturers' Institue: Palm Coast, FL, USA, 2009; pp. 289–315.
24. Vainio, A.M.; Valtonen, M.; Vanhala, J. Proactive fuzzy control and adaptation methods for smart homes. *IEEE Intell. Syst.* **2008**, *23*, 42–49.
25. Herczeg, M. The smart, the intelligent and the wise: Roles and values of interactive technologies. In Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia, Allahabad, India, 27–30 December 2010; ACM: New York, NY, USA, 2010; pp. 17–26.
26. Knox, W.B.; Stone, P. Interactively shaping agents via human reinforcement: The TAMER framework. In Proceedings of the Fifth International Conference on Knowledge Capture, K-CAP '09, Redondo Beach, CA, USA, 1–4 September 2009; ACM: New York, NY, USA, 2009; pp. 9–16.
27. Knox, W.B.; Stone, P.; Breazeal, C. Training a robot via human feedback: A case study. In *Social Robotics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 460–470.
28. Akrour, R.; Schoenauer, M.; Sebag, M. APRIL: Active Preference-Learning Based Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7524, pp. 116–131.
29. Nick, M. Reducing the Case acquisition and maintenance bottleneck with user-feedback-driven case base maintenance. In Proceedings of the FLAIRS Conference, Melbourne Beach, FL, USA, 11–13 May 2006; pp. 376–382.
30. Nick, M.; Becker, M. A hybrid approach to intelligent living assistance. In Proceedings of the 7th International Conference on Hybrid Intelligent Systems, HIS 2007, Kaiserlautern, Germany, 17–19 September 2007.
31. Khalili, A.; Wu, C.; Aghajan, H. Autonomous learning of user's preference of music and light services in smart home applications. In Proceedings of the Behavior Monitoring and Interpretation Workshop at German AI Conference, Paderborn, Germany, 15–18 September 2009.
32. Gil, M.; Pelechano, V. Exploiting user feedback for adapting mobile interaction obtrusiveness. In *Ubiquitous Computing and Ambient Intelligence*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 274–281.
33. Rashidi, P.; Cook, D.J. Keeping the resident in the loop: Adapting the smart home to the user. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2009**, *39*, 949–959.
34. ZigBee. Available online: https://en.wikipedia.org/wiki/ZigBee (accessed on 14 June 2016).
35. Cleobee. Available online: http://www.cleode.fr/en/produits.php?page=cleobee (accessed on 14 June 2016).
36. Hockley, W.E. Analysis of response time distributions in the study of cognitive processes. *J. Exp. Psychol. Learn. Mem. Cognit.* **1984**, *10*, 598–615.
37. Fleury, A.; Vacher, M.; Portet, F.; Chahuara, P.; Noury, N. A french corpus of audio and multimodal interactions in a health smart home. *J. Multimodal User Interfaces* **2013**, *7*, 93–109.
38. Vacher, M.; Lecouteux, B.; Chahuara, P.; Portet, F.; Meillon, B.; Bonnefond, N. The Sweet-Home speech and multimodal corpus for home automation interaction. In Proceedings of the 9th Edition of the Language Resources and Evaluation Conference (LREC), Reykjavik, Iceland, 26–31 May 2014; pp. 4499–4506.

39. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artif. Intell.* **1998**, *101*, 99–134.
40. Loftin, R.; MacGlashan, J.; Peng, B.; Taylor, M.E.; Littman, M.L.; Huang, J.; Roberts, D.L. A strategy-aware technique for learning behaviors from discrete human feedback. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-14), Québec City, QC, Canada, 27–31 July 2014.
41. Bellman, R. A Markovian Decision Process. *Indiana Univ. Math. J.* **1957**, *6*, 679–684.
42. Howard, R.A. *Dynamic Programming and Markov Processes*; The MIT Press: Cambridge, MA, USA, 1960.
43. Karami, A.B.; Sehaba, K.; Encelle, B. Adaptive artificial companions learning from users' feedback. *Adapt. Behav.* **2016**, *24*, 69–86.
44. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 2nd ed.; Pearson Education: Upper Saddle River, NJ, USA, 2003.
45. Langford, J.; Zhang, T. The epoch-greedy algorithm for multi-armed bandits with side information. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2008; pp. 817–824.
46. Isbell, C.L., Jr.; Kearns, M.; Singh, S.; Shelton, C.R.; Stone, P.; Kormann, D. Cobot in LambdaMOO: An adaptive social statistics agent. *Auton. Agents Multi-Agent Syst.* **2006**, *13*, 327–354.