A framework for service robots in smart home: an efficient solution for domestic healthcare

Nathan Ramoly, Amel Bouzeghoub, Béatrice Finance

A Framework for Service Robots in Smart Home: an Efficient Solution for Domestic Healthcare

Nathan Ramoly[1], Amel Bouzeghoub[1], Béatrice Finance[2]
[1]SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, Evry, France
[2]DAVID, Université de Versailles Saint-Quentin-en-Yvelines, Versailles, France
nathan.ramoly@telecom-sudparis.eu

Abstract -

Domestic healthcare is becoming more and more important as the population is growing older. The last technological progresses enable numerous possibilities for monitoring and helping users in their everyday lives. Robotics and smart home are two distinct examples. They both provide great features and possibilities, but also limits. This work addresses the combination of robots and smart home. In this paper, we present a robotic framework that relies on the smart environments strengths. We tackled numerous challenges encountered by the robot for perceiving, reasoning and acting at home and that are critical for healthcare applications. Consequently, multiple solutions are presented and evaluated through both simulation and physical tests.

Keywords: service robots, smart home, healthcare, task planner, context acquisition, ontology, activity recognition.

I. INTRODUCTION

As population is growing older, the need for domestic healthcare increases every day. Elderly people are faced with numerous health issues and need to be accompanied. On top of that, loneliness and isolation are major problems for them. Nowadays, we observe the emergence of both service robotics and smart environments towards domestic healthcare. On the one hand, robots are becoming both smarter and cheaper. Multiple projects, such as Robot-Era [1] or CompanionAble [2], are pressing forward and the first affordable personal robots, such as Buddy (http://www.bluefrogrobotics.com/) or Zenbo (https://zenbo.asus.com/) are currently being released to the public. These robots have three purposes: keep company with the user, thus being social towards him/her; monitor the user and provide feedbacks to the medical staff; and help the user by acting. Robots can efficiently act and interact thanks to their actuators, mobility and speech, but their perception is limited to their sensors, reducing their ability to monitor. On the other hand, smart environments, in particular smart homes, are rising alongside the Internet of Thing (IoT). More and more devices are now embeded with intelligence and are able to interact over the Internet or local networks. The market is rich in devices and solutions. Similarly as robots, multiple research projects and platforms are studying smart home for domestic healthcare, Mobile-Mii (http://fedev.universite-paris-saclay.fr/platforms/plateforme-dexperimentation-mobile-mii-du-cea-list) is a good example. These environments aim to monitor the user and, if possible, to act through signals or actuators. As they can rely on various spread sensors, smart environments can efficiently monitor the user. However, such systems can only perform simple tasks and has low interaction with the user. All in all, both service robots and smart homes tackle more or less the same problems, each having their pros and cons. By combining both solutions, it seems we can improve the overall quality of service: each approach overcomes the limitation of the other. For instance, a smart environment can efficiently monitor the user while a robot can interact with the user, do various tasks or gather information unreachable for the environment thanks to its mobility. However, such an interaction may not be trivial and some issues are to tackle. In this paper, we propose a framework for robots operating in smart homes. This framework aims to provide tools that are suitable for robots/smart environment interactions. Our solution aims to be environment-independent, meaning it can work with any, and without, smart environments. Moreover, it relies on three steps: perception, cognition and action. For each of them, challenges are identified, studied and solved from the perspective of the robots/smart environment interactions. This paper presents those challenges and the proposed solutions. It is structured as follows. Section II reviews the related works of robots in smart environment. We present our framework and the problematic in Section III before describing the solution in Section IV. Section V describes our experiments and results. Finally, a conclusion ends the paper through Section VI.

II. RELATED WORKS

We review some general approaches, scenarios and technologies combining robotics and smart environments. From a robot's eye, a smart environment is an opportunity; it offers new possibilities to observe, most of the time in a reliable way, and to act on the world. Be aware that sensing and acting are challenging issues in robotics. That's why such an interaction has been successfully tested in museums and homes, which are non-friendly robot environment. Shiomi & al designed a system based on robots and a smart environment to enhance the Museum of Science of Osaka [3], in this case, smart devices are mainly used for localization and users identification. They are several examples of applications at home. Baeg & al [4] added intelligence to furniture, such as tables or shelves, in order to enhance robot perception and

localization. The PEIS project [5, 6] is with no doubt the greatest effort in that direction. In this work, robots are part of the smart environment, which not only contains sensors, but also multiples actuators. In that case, smart environments and robots are one, and help the users in their everyday life. More recently, Robot-Era project, which aims to provide personal robots for elderly person, also deal with smart environments, this is not the main objective of the project and it relies on PEIS contributions. Lastly, CompanionAble [2, 11] is a project that aims to provide cheap robots to help elders by using a smart environment. The smart environment is made from multiple kinds of devices. The most common technology in this case is RFID (Radio Frequency IDentifier). RFID [3, 4, 9] tags are a cheap and efficient way to store data in the environment, they are passive (no energy needed) and easy to set up. RFID readers allow precision and/or range, according to the needs: from tag carpet [9] to beacons, the possibilities are broad. Cameras are also important for pervasive environment, even if they rely on powerful algorithm; they provide multiple information, in particular localization [3, 4, 5]. There are multiple types of cameras, infra-red or 3D cameras are great example of non classic cameras. Landmarks are also sometimes used, QRCode (or similar technology) are very convenient to locate precisely. Of course, specific sensors are not to exclude, light sensors or thermometer are quite useful at home. Actuators are not that frequent, but they do exist. They are no specific technology for this, but with the rise of connected devices, we can easily foresee a robot controlling a heater, a cellphone, a door [8] or even an armed fridge, as exposed in PEIS works. Communication is of course an issue, in most case, it is transparently handled using various technologies such as ZigBee, Wi-Fi, Bluetooth, etc… But interaction with other devices does not limit to home or museum. With the emergence of the Internet of thing, a new dimension has appeared. As Jong-Hwan Kim states [7], the next generation robots will be part of the Internet, able to virtually move and to use device through the Internet of Thing. These exist called ‘ubiquitous robot’, they follows personal robots (2nd and current generation) and industrial robot. There are already some works on ubiquitous robots. UNR-PF [10] (Ubiquitous Network Robot Platform) is a global system that allows to provide help to the user everywhere, from home, to the supermarket, by using robots, smartphone, sensor networks, etc… It relies on services and multiples layers. The LIREC project (http://lirec.eu/project) is another example where a robot is able to switch between physical reality and virtual space. Nevertheless, ubiquitous robots are still a concept and researchers are currently mostly focusing on personal robots.

Two types of architecture can be encountered: centralized or distributed. In a centralized approach [3, 4], the ‘intelligence’ and the main processes are located on one dedicated server. It can be the robot as well as another device. In this approach, the central server gathers all data from sensors and decides what to do through actuators and robots. On the other hand, in a distributed system [5], data are shared and decisions are taken via ‘deliberation’ between devices. Smart resources (sensors and actuators) directly communicate with each other, robots are also handled as devices (at least in PEIS approach). Both approaches have their pros and cons. Centralization allows to have all the data at the same place, simplifying reasoning. Also, having a dedicated device may increase the efficiency and the response time of the reasoning (more powerful, specialized). However, if it falls, all the system follows… This is not an issue in distributed approaches. This way offers a greater flexibility and robustness (a device can go out of order without impacting the whole system), however, setting up is harder and data reasoning and management is not as powerful.

In our work, as we want our solution to be environment independent, we opt for a centralized approach. Unlike the PEIS project, the main processes will be embedded in the robot. Furthermore, most of other approaches rely on a limited number of sensors or focus on a few particular types. This is a strong limitation; in consequence, we want our framework to be generic and environment independent.

III. FRAMEWORK ARCHITECTURE AND CHALLENGES

In this section, we describe the global structure of our framework. As previously explained, it relies on three steps: perception, cognition and action:

- **Perception**: the robot gathers the current context data through various sources.
- **Cognition**: the robot analyzes the context data and reasons to understand what is happening in the environment and takes a decision.
- **Action**: the robot acts on the environment to achieve some goals determined by a previous decision.

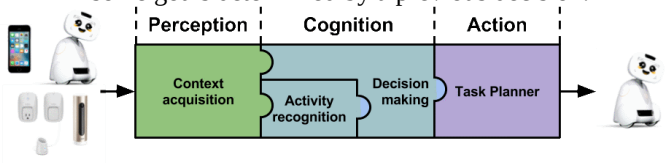

Figure 1. Framework architecture

The global process is described in Figure 1. For each step, we provide one or multiple tools that solve particular challenges. We will now identify the challenges for each step.

A. *Perception*

The first step on any robotic process is to perceive the environment through sensors. In our case, it consists of acquiring context data from various sensors, from the robot or the smart environment. The objective is to gather all the data and maintain a context knowledge that will later be used by the cognition layer.

In everyday life environment, including smart homes, uncertainty is a serious issue. In fact, we should expect that sensors will provide non-perfect data. For instance, a thermometer may not be accurate or a motion sensor may be intempestively triggered. In such cases, uncertainty can lead to the production of uncertain context data. Subsequently, this may cause problems or miscomputations in the cognition layer. For instance, if a sensor implies that the user is in room A

while he is in room B, a wrong activity may be recognized. As expressed by Ye el al. [12], there are multiple dimension of uncertainty. In a context of a robot operating in a smart home we identified four essential dimensions (depicted in Figure 2):

- **Freshness**: data is outdated. For example, if a sensor sends an event asserting the user is in room while he has just left for room B.
- **Precision**: data is correct yet inexact. For instance, a motion sensor only detects a motion, it is imprecise compared to a camera that detects motion and identifies a user.
- **Accuracy**: data is wrong or partially wrong. When the user is perceived in a room while being in another for example.
- **Contradiction**: two pieces of data provide contradictory information. For example, when the user is between two rooms and detected by sensors of both rooms.

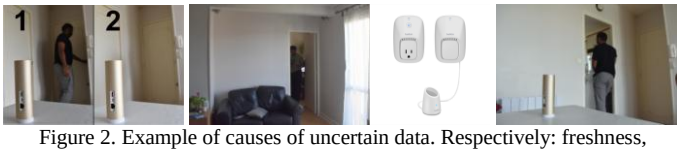

Figure 2. Example of causes of uncertain data. Respectively: freshness, accuracy, precision and contradiction

Thus, we need a tool that is able to tackle these uncertainty dimension and model context data accordingly in order to be compliant with the cognition layer. The proposed solution is described in Section IV-A.

B. Cognition

Once the context is acquired, the cognition step comes into play. It aims to analyze the context to understand what is happening and take a decision accordingly. Let us have the example of a user fall, in that case, from the acquired context data, the robot understands that the user has fallen and takes the decision to go to assist the user and/or alert the medical staff. We propose two processes for cognition: activity recognition and decision making.

Activity recognition aims to identify what the user is doing, willingly or not. 'Cooking', 'Sleeping' and 'Falling' are possible activities. The robot is able to recognize what the user is doing from vision [13, 14]. However, these image based techniques can be a bit inaccurate and require a lot of training. In fact, the main limitation is the confusion between activities: some activities imply doing similar gesture and it is sometimes hard to distinguish. Figure 3 shows such an example. But in our context, the robot's camera is obviously not the only sensor and various context data can help the activity recognition process. We propose an enhancement to a vision based activity recognition process that adjusts the recognition's result according to the context knowledge.


Figure 3. Video sample acquired by the robot. Confusion example: is the user drinking or phoning ?

Furthermore, non-vision based activity recognition techniques do exist. However a few of them are able to take into account the uncertainty (modeled in the perception layer in our case). An uncertainty friendly activity recognizer could provide more accurate results than standard approaches [15]. However, as our framework is robot-oriented, we will focus on a vision based solution in this work.

The decision-making process, as its name implies, aims to have the robot taking a decision. The idea is to identify situations that require the robot intervention, a user fall for instance. Such a tool uses the context knowledge as well as the detected activities.

Proposed tools for cognition are presented in Section IV-B.

C. Action

The action layer aims to intervene on the environment in order to reach a goal that was determined by the cognition layer. This step relies on a task planner. A task planner is a tool that generates a sequence of action (or task) to reach a given goal in a given context. Automated planning is a broad topic that goes beyond robotics. Among famous planner we can cite STRIPS (STanford Research Institute Problem Solver) [16] or HTN (Hierarchical Task Network) [17, 18, 31], these two are pioneers and leaded to numerous variations. Task planning is essential for robots, in particular in non dedicated environments such as homes. Indeed, it allows to figure out how to achieve a goal in a particular context. A classical task planner uses an input the context and a goal, and outputs a sequence of actions, in other words the plan. The planner only focuses on generating the plan and does not manage the plan execution. The plan does not change during execution. In case of a failure of an action, the planner is called again with an updated context. Failure is to be prevented, as it is time consuming due to the further actions required to reach the goal.

In our context, the robot is not the only one acting on the environment: as the user is living his/her life, he/she continuously changes the context. This variability is a problem for plan execution. In fact, for a robot, executing a plan is a matter of numerous minutes, and a lot of changes can occur within this time. And some changes can lead the plan to be outdated. For example, if the robot is asked to fetch some medications and the user moves them while it is reaching their location, it won't be able to find them, thus failing to achieve its goal. Another problem is the availability of data: in such environments, we can't ensure the robot has access to the whole context when planning. Let us have the example of fetching again; the robot knows medications can be in the kitchen on the table or in the drawer, but not their exact position, thus it doesn't know what option to select in its plan.

However, it could get the information when it reaches the kitchen, by using its cameras or reaching a sensor with a limited range for instance. By planning while not having all the required data, the robot may be running an incorrect plan. Both context variability and missing data at planning may lead to failure situations that would drastically slow down the robot. This is not acceptable as the user health can be in the balance. For instance in the case of a fall of the user, the robot shall not waste time trying possible options to reach him/her.
To solve these two challenges, we propose an upgrade of the HTN planner that combines planning and execution to enable flexibility and reactivity to the robot's plan. It is addressed in section IV-C.

IV. CONTRIBUTIONS

As expressed in the previous section, each step has specific challenges for our context of a robot operating in a smart home. Consequently, specific solutions must be proposed to each of these challenges in order to provide an efficient framework.

A. Perception

As expressed in Section III-A, context acquisition is facing four types of uncertain data: freshness, accuracy, imprecision and contradiction. We propose a tool that combines Complex Event Processing (CEP), semantic knowledge and fuzzy logic. Our solution is a three step process as depicted in figure 4.

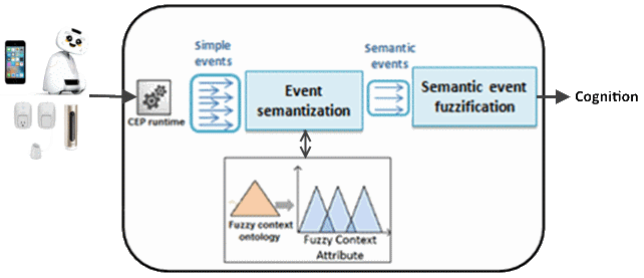

Figure 4. Context acquisition process

1. **Event Processing**: raw events acquisition from sensors, batching and filtering. Relies on a CEP.
2. **Semantization**: enhancing events with semantic knowledge and format.
3. **Fuzzification**: computation of fuzzy values by analyzing batches of events and generation of one Fuzzy Semantic Complex Event (FSCE).

The resulted FSCE are sent to the cognition layer tools that will store and use them. We will now explain each step of the context acquisition process and how they tackle the uncertainty. For a more detailed explanation, please refer to [19].

1) Event Processing

The aim of the event processing step is to gather events from various sources filter them on given criteria and batch the event in time-window. It relies on a classical CEP system, such as Coral8 [21] or, in our case, ESPER [20]. A CEP is a tool that gathers event and generate higher level one. In our work, we use it for collecting data while the computation of higher level event (FSCE) is done by the two next steps. In order to gather, filter and batch, the CEP relies on designer-defined query. Let us have an example with the following rule: *Select events of type="location" in time_batch(1min) where source = "motionSensor1" or source="camera3"*. This query gathers the event of type 'location', filters by their sources and batch them in 1 minute time-windows. The filtering allows doing a first basic handling on events. The time batching is the key feature to overcome the **freshness** issue. In fact, by considering only the events in a given time window, we prevent using outdated event. Of course, the shorter is the time-window the lower outdated events there is, but oppositely, having too short one can lead to have nearly empty batches, limiting the fuzzification process. Determining the time windows size is the matter of the nature of events and the choice is left to the designer. This step outputs a batch a filtered events and is provided to the semantization process.

2) Semantization

Based on a background knowledge represented as an ontology, the semantization phase enriches and formats the events provided in the batch.
The enrichment has two purposes. First, it enriches event data that may be weak by itself. The idea is to directly tackle the **imprecision** issue. In fact, as events' precision can be enhanced through further information stored in the expert-provided background. For example, a motion sensor can be linked to a given room and a particular user. Based on this knowledge, the semantization phase will enrich this sensor's events so that they can carry the information of the user location. The second purpose of the enrichment is to add confidence values on the event based on their parent sensor. In the background knowledge, each sensor is associated with a confidence value (provided by an expert) that will be attached to the events they generate. This confidence value aims to represent and tackle the **accuracy** of sensor.
In order to carry these further data, the events are formatted into RDF (Resource Description Framework) graphs. In other words there are formatted in an ontological format. The resulting events carry enriched information presented as object linked to their properties, as depicted in Figure 5.

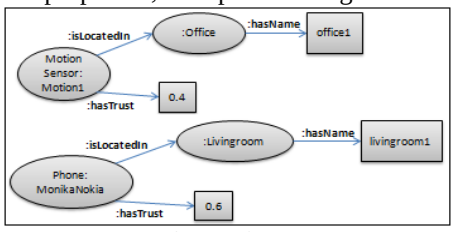

Figure 5. Example of events formatted as RDF graphs

The structure of these graphs is computed from the background knowledge. By formatting the event as RDF graph, the data are interoperable and can be easily integrated to other system, in particular cognition's tools in our case.

The semantization outputs a batch of semantic events that are passed to the fuzzification phase

.

3) Fuzzyfication

The fuzzification phase analyses a batch of semantic events to generate one Fuzzy Semantic Complex Event (FSCE). It actually aggregates all possible events with trust weights associated to each possible value. For example, as for location, the fuzzification can generate such an event: $FSCE_{location}$ = *{[livingroom; 0.33] , [office; 0.67] }*. By doing so, our system overcomes the problem of contradiction: it provides a trust value for each value and it doesn't exclude any possible value, meaning all possibilities will be taken into account in the cognition layer.

In order to compute these trust weights, the process uses a dedicated membership function on all events of the provided batch. This function uses the semantic information carried by event and computes a weigh accordingly. In our case, we use a weighted mean on confidence value, but other functions are possible for the designer.

The process then generates one single FSCE as a RDF graph with the computed trust weights modeled as annotation, as depicted in Figure 6.

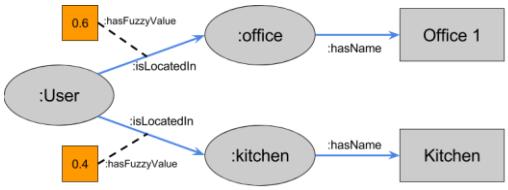

Figure 6. Example of FSCE

The generated FSCE is then provided to the cognition layer. Be aware that the context acquisition can be instantiated multiple times, each with its own query. Thus, the cognition layer is fed from various FSCE.

B. Cognition

Once the context is acquired and provided as multiple FSCE, the cognition step comes in play. It is divided in two tools: an enhanced vision based activity recognition system, and a decision maker.

1) Activity recognition

The robot has the ability to recognize the activities of the user. But, as expressed in Section III, it encounters difficulties. We enhance the vision process described presented in [13, 14]. In brief, it works following four steps that rely on several supervised learning algorithm:

1. **DenseTrack** is an algorithm that analyses an image steam to provide pixel trajectories (tracking);
2. **K-Means** algorithm is used to group trajectories into a fixed set of prototypes or clusters.
3. **Bag of words** algorithm is used to compute the frequency of 'visual words' thanks to the clusters generated by K-means
4. **Support Vector Machine (SVM),** the bags are classified using a trained SVM.

It outputs a list of activities associated with a probability. For example: *{(eating, 0.05),(phoning,0.1),(walking,0.01,…)}*. This approach was proven efficient on benchmarks, but is yet to be improved in realistic configuration, such as ours.

To improve this process, we complete it with the available context knowledge. First, the context knowledge is stored and maintained in an ontology that is fed from the perception layer output. In other word, it carries data from the FSCE, including the trust weights. Whenever the robot applies its vision algorithm for activity recognition, it checks the context knowledge and enhances its results based on simple rules. Let us have some example:

- If the phone's inertial unit exceeds a given threshold- increase "phoning" probability,
- If a movement was detected through a motion sensor located in a room with a TV – increase "remote controlling" probability,
- If the opening door sensor was triggered – increase "opening door" probability.

The probability increase takes into account the trust weights of data, but how the activity's probability is increased is defined by the designer. With such adjustments, the activity recognition is more accurate.

But the background knowledge can also refine the result of the vision algorithm. In fact, it is possible to detect more precise activities by combining the vision results and the context data. For example, if the user is 'sitting' in the kitchen around noon, the robot can infer he/she is eating. To do so, we apply reasoning rules. Here is an example written using Jena: *(?act is-a Activity) (?act a label "sitting") (?act a proba ?prob) (?anyact a proba ?anyprob) greaterThan(?prob ?anyprob) (?usr located-in ?room) (?room is-a RoommMeal) (?meal time meal ?time) equal("cur time" ?time) → (?newAct is-a Activity) (?newAct a label "eating")*.

By enhancing the vision based activity recognition with semantic background knowledge, we improve the accuracy of the results as well as their precisions. We conducted multiple experiments that are addressed in Section V. The output is the same as the vision process, namely a list of activity associated with refined probabilities. These activity are then use for decision making.

2) Decision making

The decision making process aims to determine if the robot should intervene and what it is supposed to do. It uses the results of both perception and activity recognition.

In order to decide, the robot relies on rules. The rules are provided by the designer and define the reaction the robot shall have when detecting particular context data (including activities). For example, if the user is sleeping while a window is open, the robot should decide to close the window: *(?act is-a Activity) (?act a label "sleeping")(?win is-a Window) (?win status Open) (?win is-located-in ?room) (?usr located-in ?room) → set(?win status Close).* These rules have the feature

of taking into account the trust weights of data and probability of activity by computing a confidence value for the implicated decision. This confidence value is computed by doing the mean of all trust weights, and probabilities for activities, of context data represented in the rule. After evaluation, multiple rules may be valid, in such case, the decision with the highest confidence value is selected.
We plan to use a more efficient and adaptable solution that is able to detect anomalies that are not necessarily statically given by the user. Such a work is however out of the scope of this paper, but is presented in this [22].
In any case, this layer selects a goal to achieve in order to serve the user and provide it to the action step.

C. Action

Once it has taken its decision, the robot has a goal to achieve. It now has to determine how to achieve this. To do so it relies on a task planner. In our work we selected and improved HTN (Hierarchical Task Network). We will first discuss HTN before presenting our contribution.

1) HTN

HTN is a common planning technique that relies on decomposing tasks into subtasks and on the knowledge of a hierarchy of tasks. It has various implementation [23, 24, 25, 26] and as widely in robotics for various applications [27, 28, 29, 30]. We will now present a brief description of HTN.
HTN aims to find a solution to a planning problem by decomposing tasks into subtasks. HTN relies on two types of tasks: primitive tasks and compound tasks. Compound tasks are realized by subtasks, while primitive tasks are 'ready-to-run' non-decomposable tasks. The result of the planning process is called solution, and is a totally ordered set of primitive tasks. A planning problem is defined as a 3-tuples (g, s, D), where g is a sequence of tasks to achieve, i.e. a 'goal', s is the initial state of the environment as predicates, and D is the domain. The domain D is a pair D = (O, M), where O is a set of operators and M a set of methods. An operator is a concrete executable action and it solves a primitive task. A method indicates how to decompose a compound task into a partially ordered set of subtasks, primitive or compound. It is defined by: its name, the task it 'realizes', preconditions as predicates describing when it is applicable, and a sequence of primitive or compound tasks, i.e. the subtasks. To represent the world state and preconditions, HTN relies on predicates. A predicate represents a property of the current context (see definition below) and respects the following pattern: property (id1, id2 ...). To generate a solution, HTN applies methods from the domain D to compound tasks depth-firstly, starting by goal tasks from g. For a more detailed overview of HTN please refer to dedicated works [31].
Figure 7 provides a very simple example of a HTN for a robot to go to its recharge station. In reality, HTN are obviously much more complex.

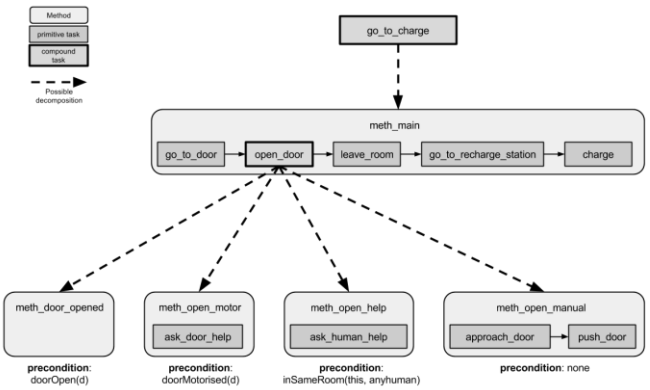

Figure 7. Simple example of a HTN

2) Dynamic HTN (DHTN)

Although HTN has several advantages and is consequently common in robotics, it is subjected to the same challenges as any task planner: in our case, the dynamism of the environment and the unavailability of some data when planning. To tackle these problems, we propose an extension of HTN, Dynamic HTN (DHTN), featuring a combination of planning and execution: instead of generating the whole plan at once, DHTN incrementally build the plan while executing it. By doing so, DHTN is able to take into account fresh context data and proposes up-to-date plans.
DHTN mainly uses the same formalism as HTN. But as DHTN also handles the execution of the plan, the notion of *status* was added to the tasks, operators and methods. There are three states: SUCCESS, FAILURE and RUNNING. An operator is RUNNING when the robot is doing the action, and its status is set to SUCCESS or FAILURE according to the actual execution results. For instance, if the robot falls while grabbing an object, the operator it was running fails. Consequently, the other elements of HTN are impacted. A primitive task succeed when one of its operator succeed and fails if not. A method fails if at least one of its subtasks has a FAILURE status and succeeds when all of its subtasks have SUCCESS status. A compound task fails if all its methods failed and succeed if at least one method did. These statuses allow DHTN to monitor the plan execution and to use this information for planning. With these features, DHTN can efficiently find an alternative plan in case of a task failure.
We proposed a novel algorithm for DHTN. Starting from an initial task, DHTN operates has follows:

- If the current task is a primitive task: DHTN selects a matching operator and the robot executes it. It waits for the operation's result. If the operator succeeds, the task is a SUCCESS and DHTN moves to the next one and reapply the algorithm. If the operation fails, the task is a FAILURE, thus its parent method and compound task also fail. DHTN then backtracks to the parent compound task and try to find another method. If there is no parent task, DHTN fails to achieve the goal.

- If the current task is a compound task: DHTN compares the current task's methods' preconditions to the current context and selects a compatible method. By doing so, DHTN takes into account the last changes in the environment and selects an up-to-date sub-plan. Furthermore, by selecting a method when required, the robot can access data it could have missed if it would have planned beforehand. DHTN then moves to the first subtask of the selected method and reapply the algorithm.

By combining execution and planning, DHTN is able to manage a reactive and flexible plan. It limits the number of failures and, in case it couldn't be prevented, finds alternative solution without launching the whole process again. Experiments underlining this strength are presented in the next Section. For a more complete work about DHTN, please refer to [30].

V. EXPERIMENTS AND EVALUATIONS

Our framework was implemented and tested through both simulation and physical tests.

For simulation, we rely on Freedomotic (http://www.freedomotic.com/), an open source development framework to manage smart spaces, in particular smart homes. Freedomotic allows todefine data sources, actuators and carries a simple event processing system. The represented devices can be linked with real one or simulated one: by doing so, it is easy to implement in real life a system tested in the framework. One of Freedomotic main feature is its compliance with plug-in: we can easily find any plug-in we might need thanks to a play-store like tool and we can create our own upgrades. Figure 8 displays a screenshot of Freedomotic interface.

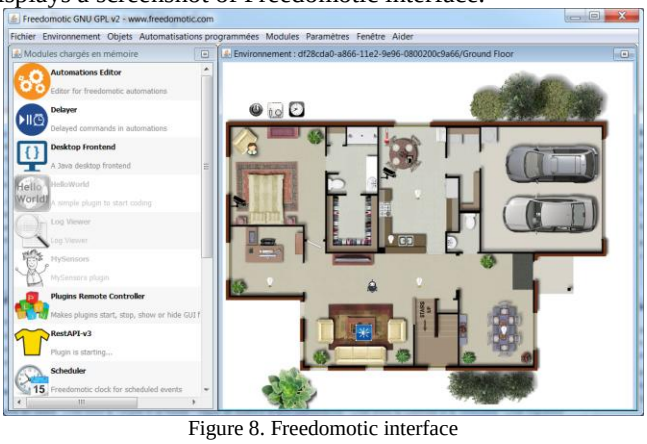

Figure 8. Freedomotic interface

Furthermore, we also tested our contributions with a robot and a smart environment. We used a Nao H25 robot [32], it is a popular humanoid robot that has a lot of abilities, such as locomotion or speech. As for the smart environment, we relied on the Hadaptic (http://hadaptic.telecom-sudparis.eu/fr/introduction/) platform. The platform is composed by a modular room, furniture and various sensors, such as motion sensors, opening sensors, thermometers and beacons.

Figure 9. The Nao robot and the Hadaptic platform

Each of our contributions was tested and evaluated using Freedomotic or the Hadaptic platform. We will now review our results.

A. *Perception*

We tested our context acquisition tools through simulation. We sat up a simulation environment based on Freedomotic that enables uncertain data generation. Our experiments aimed to show that our system was able to improve an activity recognition process such as [15] by tackling and modeling the uncertainty. We implemented and simulated a scenario and compared the correctness of an activity recognition system with and without our solution. Results can be found in Figure 10.

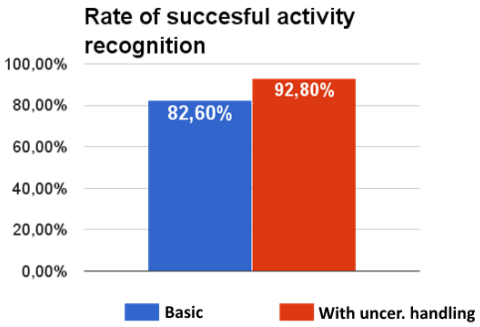

Figure 10. Activity recognition rate improvement

In Figure 10, the blue column represents the activity recognition rate with a basic context acquisition while the red one shows the results with our uncertainty aware solution. With our solution, we improved the rate of successful recognition by 10%, which is a significant gain, in particular for healthcare applications. This proves our contribution to be pertinent.

B. *Cognition*

In order to evaluate the interest of combining a vision based activity recognition process with further context data, we conducted experiments on several test subjects for three scenarios. We selected three activities (scenarios) to test: remote controlling a TV, opening a door, and phoning. A test run consists in a person doing a gesture in front of a Nao robot as showed in Figure 11. The robot acquires a video sample and applies the vision algorithm to identify the activity. It then adjusts the results based on a context knowledge provided by the smart environment through sensors including smart phones and opening sensor. We conducted our tests on 12 subjects.

Each one was asked to repeat 10 times each gesture: each scenario had 120 runs. We then compared the result of the vision only recognition and the enhanced one.

Figure 11. Example of an activity recognition test run

Results are displayed in Figure 12. The blue column shows the vision only correct recognition rate will the red one shows the improved results. The vision only process was fully confronted to the issue of confusion. Indeed, the activity probability distribution outputted was very narrow. In other word, the vision recognition process was not far from the result, yet couldn't allow clearly identifying an activity. In consequences, even a small adjustment with context data allows to clarify the recognition. This explains the huge difference between the two techniques. Nevertheless, it proves that such a combination is pertinent and efficient. You can find more detailed results in [33].

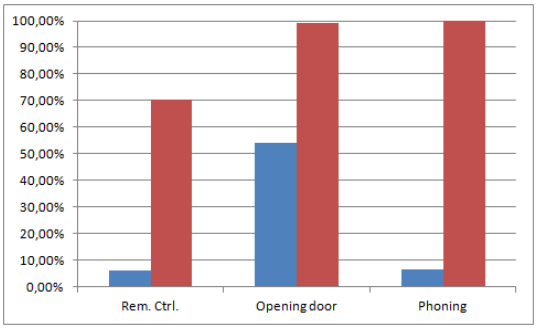

Figure 12. Correct activity recognition rate vision only (blue) and with context data enhancement (red)

C. *Action*

DHTN was implemented and tested through both simulation and a physical robot. Some proof of concept tests were conducted with Nao, as displayed in Figure 13 that depicts the reactivity of DHTN. Videos are also available online: http://nara.wp.tem-tsp.eu/what-is-my-work-about/dynamic-hierarchical-task-network-dhtn/.

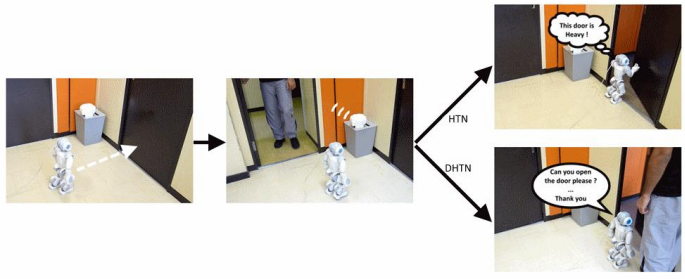

Figure 13. Action layer tests with a Nao robot

In order to evaluate it, we again used Freedomotic and designed a simulation environment that simulates a dynamic environment. We measured the number of tasks required to reach an objective for various level of dynamism and compared DHTN to HTN. Results are displayed in Figure 14.

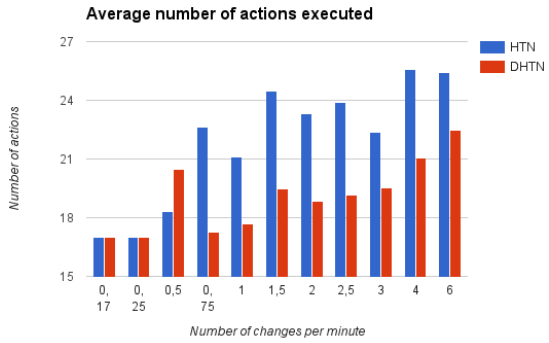

Figure 14. Number of actions required to reach a goal for various change rates

As we can see in Figure 14, DHTN needs fewer actions to reach the goal than HTN. As HTN plan is generated before running it and is not reactive, the plan often fails, leading to numerous replanning and a lot wasted task executions. DHTN on the other hand, is successfully able to take into account the last changes in the context thus preventing doing action that are meant to fails.

VI. Conclusion and Perspectives

In this paper, we presented a framework for robots operating in smart homes for healthcare domestic applications. We tackled multiple challenges including uncertainty of data and dynamism of the environment to provide and accurate and efficient robotic system. All our tools were tested and validated using both simulation and real case experiments.

In future works, we aim to improve the decision making process to handle efficiently uncertainty and priorities. We also want to test our whole framework in a real case application using the Evident platform (https://evident.telecom-sudparis.eu/).

References

[1] Bevilacqua, Roberta, et al. "Robot-era project: preliminary results on the system usability." *International Conference of Design, User Experience, and Usability*. Springer International Publishing, 2015.

[2] Schroeter, Ch, et al. "Realization and user evaluation of a companion robot for people with mild cognitive impairments." *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013.

[3] Shiomi, Masahiro, et al. "Interactive humanoid robots for a science museum."Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. ACM, 2006.

[4] Baeg, M. H., et al. "Building a smart home environment for service robots based on RFID and sensor networks." Control, Automation and Systems, 2007. ICCAS'07. International Conference on. IEEE, 2007.

[5] Saffiotti, Alessandro, and Mathias Broxvall. "PEIS ecologies: Ambient intelligence meets autonomous robotics." Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies. ACM, 2005.

[6] Broxvall, Mathias, et al. "PEIS ecology: Integrating robots into smart environments." Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on. IEEE, 2006.

[7] Jong-Hwan Kim, Yong-Duk Kim, and Kang-Hee Lee. The third generation of robotics: Ubiquitous robot. In Proc of the 2nd Int Conf on Autonomous Robots and Agents, 2004.

[8] A Graeser. Ambient intelligence and rehabilitation robots—a necessary symbiosis for robust operation in unstructured environments. In Electronics and Telecommunications (ISETC), 2010 9th International Symposium on, pages 9–16. IEEE, 2010.

[9] Ricardo Tesoriero, R Tebar, Jose A Gallud, Mar ´ ´ıa Dolores Lozano, and Victor M Ruiz Penichet. Improving location awareness in indoor spaces using rfid technology. Expert Systems with Applications, 37(1) :894–898, 2010.

[10] Kamei, Koji, et al. "Cloud networked robotics." Network, IEEE 26.3 (2012): 28-34.

[11] Gross, H., Schröter, C., Mueller, S., Volkhardt, M., Einhorn, E., Bley, A., ... & Merten, M. (2011, October). I'll keep an eye on you: Home robot companion for elderly people with cognitive impairment. In Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on (pp. 2481-2488). IEEE

[12] Ye, J., Dobson, S., McKeever, S.: Situation identification techniques in pervasive computing: A review. Pervasive Mobile Comput. 8(1), 36–66 (2012)

[13] M. A. El-Yacoubi, H. He, F. Roualdes, M. Selmi, M. Hariz, and F. Gillet, "Vision-based recognition of activities by a humanoid robot," International Journal of Advanced Robotic Systems, vol. 12, 2015.

[14] M. E.-Y. Mouna Selmi and B. Dorizzi, "A two-layer discriminative model for human activity recognition," IET Computer Vision, 2016.

[15] Hela Sfar, Amel Bouzeghoub, Nathan Ramoly and Jérôme Boudy, "AGACY Monitoring: a hybrid model for activity recognition and uncertainty handling", Extended Semantic Web Conference (ESWC) 2017

[16] R. E. Fikes, N. J. Nilsson, Strips: A new approach tothe application of theorem proving to problem solving, Artificial intelligence 2 (1972) 189–208.

[17] A. Tate, Project planning using a hierarchic non-linear planner, Department of Artificial Intelligence, University of Edinburgh, 1976.

[18] E. D. Sacerdoti, Planning in a hierarchy of abstraction spaces, Artificial intelligence 5 (1974) 115–135.

[19] Jarraya, Amina, et al. "FSCEP: A New Model for Context Perception in Smart Homes." *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer International Publishing, 2016.

[20] Event stream intelligence, E.: Espercomplex event processing (2010)

[21] Morrell, J., Vidich, S.: Complex event processing with coral8 (2007)

[22] Hela Sfar , Nathan Ramoly , Amel Bouzeghoub , and Beatrice Finance, "CAREDAS: Context and Activity Recognition Enabling Detection of Anomalous Situation", Conference on Artificial Intelligence in Medicine (AIME) 2017

[23] K. Erol, J. A. Hendler, D. S. Nau, Umcp: A sound and complete procedure for hierarchical task-network planning., in: AIPS, volume 94, pp. 249–254.

[24] K. Currie, A. Tate, O-plan: the open planning architecture, Artificial Intelligence 52 (1991) 49–86.

[25] D. Nau, Y. Cao, A. Lotem, H. Muñoz-Avila, Shop: Simple hierarchical ordered planner, in: Proceedings of the 16th international joint conference on Artificial intelligence-Volume 2, Morgan Kaufmann Publishers Inc., pp. 968–973.

[26] D. S. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, F. Yaman, Shop2: An htn planning system, J. Artif. Intell. Res.(JAIR) 20 (2003) 379–404.

[27] Weser, M., Off, D., Zhang, J.: Htn robot planning in partially observable dynamic environments. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 1505–1510. IEEE (2010)

[28] Lallement, R., De Silva, L., Alami, R.: Hatp: An htn planner for robotics. arXiv preprint arXiv:1405.5345 (2014)

[29] Milliez, G., Lallement, R., Fiore, M., Alami, R.: Using human knowledge awareness to adapt collaborative plan generation, explanation and monitoring. In: The Eleventh ACM/IEEE International Conference on Human Robot Interaction, pp. 43–50. IEEE Press (2016)

[30] Ramoly, N., Bouzeghoub, A., Finance, B.: Context-aware planning by refinement for personal robots in smart homes. In: ISR 2016: 47st International Symposium on Robotics; Proceedings of, pp. 1–8. VDE (2016)

[31] Georgievski, I., Aiello, M.: An overview of hierarchical task network planning. arXiv preprint arXiv:1403.7426 (2014)

[32] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. O. Monceaux, P. Lafourcade, B. Marnier, J. Serre, B. Maisonnier, Mechatronic design of nao humanoid, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, pp. 769–774.

[33] Nathan Ramoly, Vincent Vassout, Amel Bouzeghoub, Mounim A. El Yacoubi, Mossaab Hariz, "Refining Visual Activity Recognition with Semantic Reasoning", IEEE International Conference on Advanced Information Networking and Applications (AINA), 2017