APPLYING AUTOMATION IN REMOTE HEALTH CARE

Truong Thi Bich Thanh

The University of Danang, University of Science and Technology; ttbichthanh@gmail.com

Abstract - The increasing cost of aging population and dependants has now become a growing concern. However, the advancement of science and technology, especially of information technology, has created opportunities to improve health care services. This is also a motivation for new researches designed to supplement the capabilities of the elderly as well as the disabled to ensure that they can maintain a healthy and independent lifestyle in their own homes as long as possible. In the present context, the research in this paper presents an idea for health care services at home via an analysis of users' habits. Existing home entertainment tasks and other activities are regarded as built-in sensors. Based on the modeling of the tasks, a reinforcement algorithm is applied to the analysis of users' habits. Then frequently used scenarios supplementing user capabilities are discovered.

Key words - analysis of users' habits; scenario; event quality; grouping; modeling

1. Introduction

According to the chapter 'Population Division' in [2], in 2000, 11% of the world's elderly people aged 60 or older are 80 or more. By 2050, this rate could increase to approximately 20%. With this rapid growth rate of the elderly population, the need for services for aged and disabled people is increasing, including the need for assisted-living facilities. We also observe a trend toward maintaining people in their private homes as long as possible. This is motivated first by people's own wishes, and second by cost reduction objectives. In this context, more and more research is being done on the monitoring of dependant people (i.e. elderly and disabled people) in their own environment, with more or less intrusive approaches such as telemonitoring or sensor techniques. These techniques will allow the residents to remain safely in their home far longer than could otherwise be expected. Our work takes place in this context, and includes two steps: i) providing the user with new services based on an analysis of his habits, namely the way he is using the home automation and multimedia services; ii) providing a low-level and non-intrusive personal supervision based on the above analysis.

This paper is organized as follows: after reviewing in Section 1 the general background and our own approach, we will introduce the modeling used in our work in section 2. In section 3, we will present the analysis of our approach. Section 4 presents the test platform used for the validation of our work, and Section 5 describes the obtained simulation results. Finally, in Section 6, we will draw up some conclusions and perspectives.

1.1. Background

To determine what the elderly require, to enable them to remain in their homes as long as possible, Bargers et al. described in [3] a mixed-model framework, to develop a new probability model of behavior patterns. In the same field of research (tracking a user's behavior), other contributions are presented in [4, 5]. In terms of Smart Home, many studies target technical support for disabled and elderly people, with the design of an intelligent environment adapted to the users' needs [6, 7].

Most of these approaches integrate various sensors and cameras to most of the environment's devices. However, input from users and professionals, including occupational therapists (OT), indicate that such intrusive methods are uncomfortable and therefore not easily accepted. This is an important issue since the primary objective is the user's safety and well-being. Furthermore, the use of sensors also requires an investment in costly equipment.

1.2. Our Approach

With the aim of contributing something new to the support and assistance of dependant people, we attempted to find a non-intrusive solution, without sensors, and based on existing services. We also used our analysis to propose an online composition of services, i.e. a proactive meta-service. In our approach, using returned services, we built up an ontology model of daily services and relevant scenarios, to model the existing home automation and multimedia system. Instead of sensors, we calculated the quality of service (QoS) and probability models, both for anomaly detection and for the day by day monitoring of the user. The QoS specification of our approach is directly based on the users' needs and habits. With a modified reinforcement algorithm [9] presented in this paper (in Section 3), we can detect the user's habits and offer him new automatic scenarios. Figure 1 shows the principle of our method.

Occupational therapists (OTs) have an important role to play in the search for techniques to assist dependent persons. This point is often forgotten in existing approaches. Hence, in our approach, as real-life experience proves that cooperation with these professionals is essential, we integrate dependent persons and OTs into the loop.

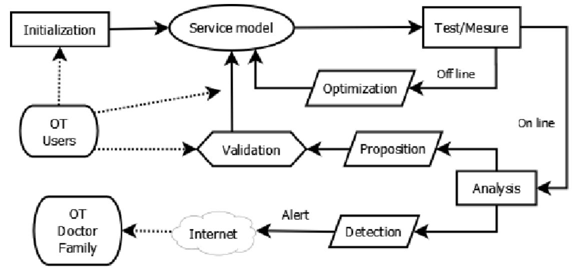

Figure 1. *Scheme of service scenario adaptation*

Our approach is based on two steps. The first is performed online. In the Initialization phase, with the help of an OT, we draw up a Service model from the existing home automation and multimedia system. In this online phase, the system's design is optimized to improve the QoS, and the QoS values adapt proposed services with the help of the OT. The second phase is run online. Following the optimization of the QoS criteria, our analysis features a modified reinforcement algorithm, in order to offer new scenarios. The "Proposition" phase is then validated by the OT and the user's opinion, and the service model updated. Adding probability models to the analysis allows us to detect an anomaly, a departure from the usual user profile, and to warn the family members, doctor and OT via internet.

2. Modeling

In the context of our subject - existing home automation and multimedia services - our approach is based on the ontology of returned services. Therefore, our first important design phase is the service and scenario modeling described in this section.

2.1. Service modeling

In order to provide semantics for the various elements of the service architecture, we will give some definitions in the context of home automation and multimedia systems.

- Operation: an operation is a function performed by a resource (e.g. 'switch on light' with a PDA, 'turn on TV' with a remote control).

- Service is a function or a set of mutually dependent functions carried out by the user.

We set for each service a Quality of Service value (QoS). We recognize two types of service:

- An elementary service is a function (e.g. 'turn off light'), or a set of mutually dependent functions (e.g. 'open door' consists of two mutually dependent functions 'command open door' and 'door open'). An elementary service cannot be broken down into sub-services.

- A scenario is made up of at least two elementary services (e.g. a 'go out' scenario is achieved through a set of services: 'open door', 'turn off light' and 'close door'.)

Within a scenario, according to the importance of function failure, we classify functions into two categories:

- Critical function: a function is critical if its failure causes the failure of the whole scenario.

- Normal function: a non-critical function.

To define the status of the services, we have three service modes:

- Out of order mode: the mode which causes the scenario's failure.

- Deteriorated mode: the mode indicating a decrease in the scenario's QoS without bringing the scenario to an end.

- Normal mode: the mode in which all functions run normally.

Each of the means by which a function can be activated is considered as a distinct operation. We therefore assume the existence of different types of resources, allowing the user to activate a service through different means.

- Direct: the user accesses the resource directly, we have a type of resource or device.

- Electronic: through electronic control buttons.

- Domotic: through a user interface such as PDA, PC or touch screen.

From these definitions, we can build up a hierarchical architecture of services, from which we can acquire the configuration of a scenario brought about by a sum of services.

2.2. Scenario graph

A set of at least two services make up a scenario; a service may contain several functions. Thus, the performance of a scenario corresponds to an ordered performance of all the functions which make up the scenario. In order to present this form of scenario, we will show the construction of a scenario graph.

Beside simple services which involve only one operation such as 'Switch on light', 'Turn on television', there are complex services made up of several functions, in which the occurrence of the next function depends on the result of the previous one. For example, in order to open a door, the function 'Unlock door' must already be accomplished. In order to draw up a scenario graph, we need to discriminate, in the scenario, between functional dependency and ordering dependency.

- Functional dependency: the term is used to express the connection between a sequence of functions performed in a predefined order. The occurrence of the next function depends on the result of the previous function in the sequence. Therefore, in order to complete this sequence, all the functions must have been executed. For example, achieving the service 'Listen to Web radio' depends on three functions with functional dependencies:

+ Go on the Internet

+ Connect to a selected site

+ Play the radio

The service 'Listen to Web radio' implies that these three functions run correctly.

- Ordered dependency: the term is used to express the connection in a sequence of normal functions in temporal order. The performance of a function does not depend on the result of a previous function. For example, we have a sequence of three functions: 'switch on light', 'open shutter', and 'turn on television', which is performed in temporal order one after the other, but the function 'open shutter' does not depend on the result of the function 'switch on light'.

With these definitions, a scenario can be presented as a functions graph as shown in Figure 2.

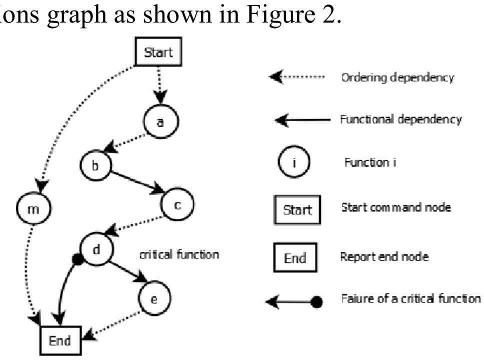

Figure 2. *Illustration of a scenario as a function graph*

In this graph, the nodes perform the functions in the scenario. The dotted edges represent the ordered dependency of two consecutive functions, whereas the plain edges denote the functional dependency between two functions. Node 'End' and node 'Start' are the graph's terminal nodes, indicating the beginning and end of the scenario. If a critical function fails, the scenario stops immediately; a point-edge to the 'End' node is realized.

This construction makes it possible to visualize all the data contained in a scenario, such as the way a scenario is performed, critical functions within the scenario, as well as the relationship between the functions. This graph therefore enables us to describe the new scenarios provided by the analysis presented in the following section.

3. Analysis

As shown in Figure 1, with the acquired service models, our analysis consists in offering new possible scenarios and detecting anomalies. To be able to offer new scenarios with a better QoS, we need to learn the user's habits. This is one of the main purposes of our analysis. The guiding principle of our work is shown in Figure 3.

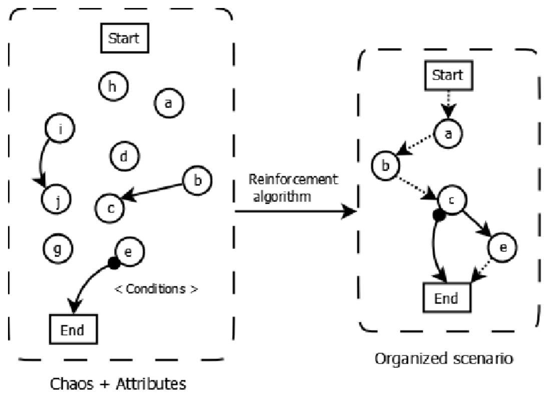

Figure 3. *Our method's guiding principle*

From the chaos of services in the user's environment, we learn the user's habits through a modified reinforcement algorithm, and then detect the sets of usual services to be offered in new scenarios.

3.1. Modified reinforcement algorithm analysis

Every day, the user performs various activities, among which can usually be detected habits, based on sets of services requested in a coherent way. It is well known that for disabled and elderly people with a limited movement capacity, it takes a long time to achieve a scenario consisting of several services, if they are performed separately. In order to reduce effort and to improve access to services, we collect the sets of services often performed together, through a reinforcement algorithm, and make them accessible within a scenario launched by a single command. Our algorithm is based on the graph construction.

- Vertex i: the service i.

- Edge: expresses the continuity of two services i and j, each edge being characterized by a weight value (i, j) which is reinforced with each repetition of the 'i, j' set.

In order to detect whether a pair of two services (i, j) occur, we use a time window T. Basically, we limit the search space to compact scenarios, namely scenarios providing a number of services in a short period of time. For example, we limit the T value to a predefined value corresponding to the user's needs - or according to the OT's opinion - (e.g. T = 30 minutes). Because the time activation between services is an important parameter in a context dealing with dependent people, the smaller the interval of time activation between services, the greater the weight of the edge. We therefore consider time intervals within the window T, in order to take into account the importance of time activation between services. The principle of this algorithm is therefore based on the computation of the weight (i, j) through the following formula:

$$W_n = \begin{cases} 1 & if\ time\ interval\ independant \\ 1 + \frac{(N-n)}{N-1} & otherwise \end{cases} \quad (1)$$

Where N is the number of time intervals in the time window T, and n is the nth interval ($1 \le n \le N$).

Then, the value of weight(i,j) updated is given by

$$weight(i,j) = weight(i,j) + W_n \quad (2)$$

Observing the above formula, it can be noted that the computed value of weight (i, j) presents an occurrence percentage for a pair of services (i, j). As a result, we obtain a graph of services in which the weight (i, j) of each pair of services (i, j), is sufficient, according to the OT's opinion (e.g. weight (i, j) >$P_{threshold}$).

On the basis of this graph, we can offer new scenarios by assembling possible sets out of services already existing in the graph. With these new scenarios, the user can access a set of services with a single command. If the user changes his habits, the reinforcement algorithm can learn the new behavior, and adapt the services to this change. Finally, the scenario graph can be used to present the obtained scenario in a time order corresponding to the performance time of the services in the scenario.

To evaluate the proposed automatic scenarios, we need to measure how satisfactory they are in relation to the user's needs and capabilities. It is moreover essential to quantify the advantages of the proposed scenarios.

3.2. QoS validation

In order to validate the performance of proposed scenarios, we use the QoS criterion to assess user satisfaction as to the performance of a service. QoS is the quality of service as perceived by the user.

In the context of home automation and multimedia systems, we take into account the user's needs as well as the user's ability to perceive the QoS. We therefore extract the models of QoS calculation according to user needs and user abilities as well as user habits. Since our calculation is directly related to the user's needs, an improved QoS value should produce an improved quality of life. In this sense, a service performed automatically, through an automatic resource, must achieve a maximum QoS value.

According to this definition of service modeling, the QoS of an operation, generated by the performance of a function j on a physical resource I, is given by:

$$QoS_{ij} = R_i \times f_j \quad (3)$$

Where: $0 \le R_i \le 1$: specific QoS resource i

$0 \le R_j \le 1$: specific QoS function j

For a service consisting of a sequence of several functions in ordered dependency, the QoS is computed with the following formula:

$$QoS = \frac{1}{n}\sum_{j=1}^{n} QoS_{ij} = Moy_j(QoS_{ij}) \quad (4)$$

At the scenario level, if a critical function fails, the scenario is interrupted and we obtain a zero value of QoS. Since the performance of a scenario depends on the operation of the critical function it contains, we calculate the QoS of a scenario with the following formula:

$$QoS = min\{Moy_j(QoS_{ij}), QoS_{critical}\} \quad (5)$$

Where

$QoS_{critical}$: QoS value of all the contained critical functions in the scenario.

If this principle is applied to new scenarios, once the user accepts our scenario proposal, all the services within the scenario are performed by automatic resources, offering a maximum QoS. Otherwise, the user must activate each service within the scenario manually, and the resulting QoS is lower than that of the automatic scenario. This difference in QoS is illustrated in Figure 4.

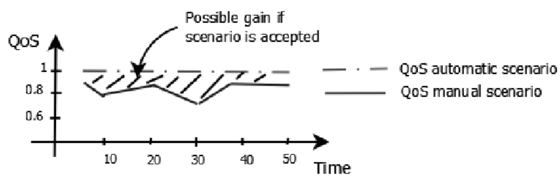

Figure 4. *The QoS difference*

The above figure shows how the better QoS of the automatic scenario generates both a gain of time and a gain of effort for the user. Therefore, the QoS validation proves the relevance of new scenarios.

In short, on the basis of returned services for the user, we can perform the analysis which enables us to create new scenarios with better QoS. Then, by observing the performance of the accepted scenarios over time, and in relation to the user's habits, we can detect possible anomalies. Without using sensors, our method shows how user habits can be monitored in a non-intrusive way, and warning signs detected. At this point, before going on to actual experimentation with the users, the relevance of our models must be assessed.

4. Test plateform

4.1. Introduction

In order to test both our model and our approach of dynamically adapting the services to the user through solutions of non-intrusive monitoring, we developed a simulator using the Scilabsoftware [8]. This is an open-source equivalent of Matlab, used to simulate the user's everyday activities. Moreover, this software enables us to create a reinforcement algorithm, and to draw up a scenario proposal graph automatically, in conjunction with the Graphvizsoftware. For these reasons, we chose a simulator for our test platform.

4.2. Simulation design

This subsection describes the principles of our simulator's design. Basically, the simulator is used to generate typical events, derived from the user's activities, and to show the QoS of the services requested. Since our method is built into existing home automation and multimedia systems, the simulator's input is the list of services including probability, dependencies, resource type and criticality of the services. These profiles, based on interviews conducted by the OT at Kerpape center [1], are imported into the user's profile data in the simulator. This simulator also has the capacity of integrating the type of dynamic analysis introduced in the previous section, to draw up better service proposals and new scenarios. This information is transmitted by internet to both the OT and the users for validation. As a result, our method can be applied to a close approximation of the user's real daily life. The principle of the simulator is shown in Figure 5.

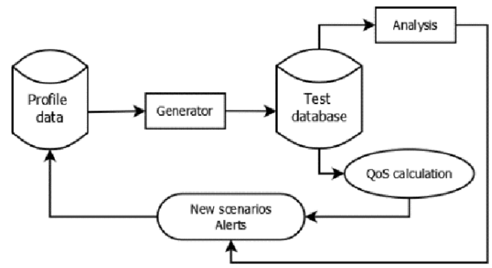

Figure 5. *Scheme of simulator design*

As can be seen in this figure, on the basis of the profile data obtained from information on the user's daily activities, a set of everyday services is generated, simulating a real-life period of N days. From this output, we obtain a test database enabling us to analyze the use of services and perform the QoS calculation. We then apply the reinforcement algorithm to the generated events to draw up our proposal for a new scenario. By observing the use of the accepted scenarios in the defined time period, we can detect warning signs in discrepancies with the user's usual habits. Finally, the user profile data is updated with the accepted scenarios. Due to the attributes of profile data based on real-life observation, the generator can build up a relevant test database. Our analysis thus provides reliable results, adapted to the user's needs.

5. Simulation results

This section describes the results of the experimental simulation. According to the simulator design diagram, the engineering of a simulation consists in the following steps:

- Step 1: Specify the table of services based on real-life observation and OT advice. For example, Figure 6 illustrates this type of table:

Events	Functions	Resources	Criticality	Probability laws
1	Fi -->Fj -->Fk	Ri -->Rj -->Rk	No	Gaussian
2	Fm -->Fn	Rm -->Rn	Yes	Poisson
3	Fp	Rp	Yes	Uniform
...	....	....	....	....
M	....	....	....	....

Figure 6. *Table of the user's everyday activities*

- Step 2: Simulation of N days based on probability. Basically, from the probability of the need for each service, we draw up the list of the daily services required by the user as shown in Figure 7.

	Time	Daily activities	Resource	E14	15:00:00	Turn on computer	PC
E1	08:00:00	Switch on light	PC	E15	17:00:00	Turn off computer	PC
E2	08:05:00	Open shutter	PC	E16	19:00:00	Switch on light	PDA
E3	08:10:00	Turn on TV	PC	E17	20:00:00	Turn on TV	PDA
E4	08:15:00	Turn on hot water	PC	E18	20:30:00	Watch DVD	PDA
E5	08:30:00	Unlock door	PC	E19	21:00:00	Turn on light ext	PDA
E6	08:45:00	Turn off TV	PDA	E20	21:15:00	Hang on telephone	PDA
E7	08:55:00	Open door	PDA	E21	21:30:00	Hang up telephone	PDA
E8	09:00:00	Switch off light	PDA	E22	21:50:00	Close shutter	PDA
E9	09:05:00	Close door	PDA	E23	22:00:00	Turn off DVD	PDA
E10	13:00:00	Open door	PDA	E24	22:00:00	Locate beb	Touch screen
E11	13:10:00	Close door	PDA	E25	22:15:00	Turn off TV	PC
E12	13:25:00	Install beb	PC	E26	22:30:00	Switch off light ext	PC
E13	14:30:00	Uninstall bed	PC	E27	22:40:00	Switch off light int	PC

Figure 7. *Table of user profile*

- Step 3: Analysis of user habits through the reinforcement algorithm and the QoS calculation.

- Step 4: Offer of new services. Based on the results of Step 3, new scenarios made up of relevant services are automatically drawn up.

- Step 5: User agreement. When the user accepts a new scenario, this means that the habits detected are reliable. Instead of having to activate all the services manually, the user can press one button to access the entire scenario. This reduces the user's effort while improving his or her access to services.

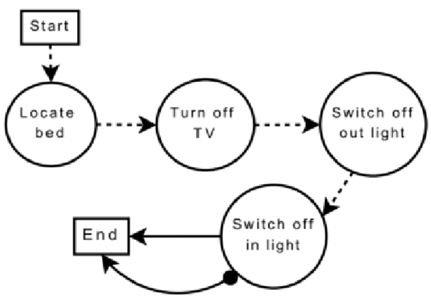

Figure 8. *Graph of "Sleep" scenario proposal*

For instance, by applying the five steps listed above with a threshold value of 50%, a 'Sleep' scenario has been obtained, consisting of a set of services rendered in a predefined order. This new scenario has been automatically drawn up, as shown in the scenario graph (Figure 8).

Our test is based on observation, and elaborated in collaboration with OTs from Kerpape Center, a large treatment center for the disabled. This figure shows a critical "Switch off light" function. The activation of the whole scenario depends on the activation of this specific function: if it operates normally, this automatic scenario gains maximum QoS. While in manual way, obtained QoSvalue is smaller due to difficulty of user in activationaction for each service. Figure 9 shows the QoS of a proposed scenario with better value.

From the simulation results, we can derive a non-intrusive observation of the user through his activities with existing home automation and multimedia systems. If the user's behavior changes, the reinforcement algorithm makes it possible to detect these new habits, and to put forward new scenarios adapted to the change.

This paper has described a non-intrusive method with a test platform in SCILAB to detect automatically the user's habits and to offer new scenarios. The result enables us to observe the user's daily life without recourse to the use of sensors, and to improve the user's quality of life while facilitating his or her use of daily services.

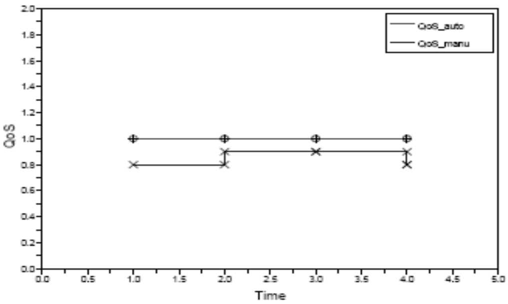

Figure 9. *QoS value of proposed scenario*

In the next step, our simulator is used to test our strategies of anomaly detection, so as to offer a complete non-intrusive monitoring of the users' daily life. To detect anomalies, a probability model for computing the duration or delay in the use of a service is given. For real-life experiments, we plan to use an open-source Linux MCE to present the user interface – a well-adapted solution to create a genuine test environment in a user's home or in one of the rooms.

TÀI LIỆU THAM KHẢO

[1] Kerpape mutualistic functional reeducation and rehabilitation center.

[2] World population ageing 1950-2050

[3] http://www.un.org/esa/population/publications/worldageing19502050/, 2002

[4] T.S.Barger, D.E.Brown, and M.Alwan. Health-status monitoring through analysis of behavioral patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(1):22–27, 2005.

[5] N.Kushwaha, M.Kim, D-Y.Kim, and W-D.Cho. An intelligent agent for ubiquitous computing environments: Smart home UT-AGENT. *In WSTFEUS, pages 157–159. IEEE Computer Society, 2004.*

[6] Dobkin, Bruce H., and Andrew Dorsch. The Promise of mHealth: Daily Activity Monitoring and Outcome Assessments by Wearable Sensors. *Neurorehabilitation and neural repair 25.9 (2011): 788–798. PMC. Web. 2 May 2016.*

[7] Ali Hussein and all. Smart Home Design for Disabled People based on Neural Networks.*Procedia Computer Science, Volume 37, 2014, Pages 117-126.*

[8] Basma M. Mohammad El-Basioniand all. Independent Living for Persons with Disabilities and Elderly People Using Smart Home Technology. *International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 3, Issue 4, April 2014.*

[9] S.Campell, J-P.Chancelier, and R.Nikoukhah. *Modelingand Simulation in Scilab/Scicos*. Hardcover, 2006.

[10] R.S. Sutton and A.G.Barto. Reinforcement Learning:An Introduction (Adaptive Computation and MachineLearning). Hardcover edition, 1999.

(The Board of Editors received the paper on 12/04/2016, its review was completed on 15/05/2016)