The Intelligent Butler: a virtual agent for disabled and elderly people assistance

Gabriel Fiol-Roig, Diana Arellano, Francisco J. Perales, Pedro Bassa and Mauro Zanlongo

Departamento de Matemáticas e Informática
Universitat de les Illes Balears
Carretera de Valldemossa, km 7.5, 07122 Palma de Mallorca, Spain
biel.fiol@uib.es;diana.arellano@uib.es;paco.perales@uib.es

Abstract. Social assistance constitutes an increasing problem in developed countries, which can be considered from two dimensions: the home and the hospital frameworks. Anyway, most of the tasks have to do with aiding people with limitations in complex environments as a hospital or a house. Intelligent agents constitute a powerful approach in designing computer systems making possible the interaction of the users (elderly and disabled people) with the elements of a domotics environment. Such a purpose can be achieved through the unification of artificial intelligence techniques, virtual reality, multimodal interfaces and digital nets with domotics services. This paper describes the design and implementation of the prototype for a virtual agent capable of attending disabled people in a home environment. The results are shown through a computer program that simulates the behaviour of the agent in developing some typical functions.

Keywords: Artificial intelligence, human-computer interaction, virtual home assistance, facial animation.

1 Introduction

Developing and integrating agent technologies in real world applications is a complex task, since not a few research and technological areas are involved. Some of these domains are distributed object technology, digital networks, 3D virtual words, virtual reality, multimodal interfaces, and artificial intelligence, among others. Such a complexity and diversity requires considering a first step where the behaviour of the agents is tested in a controlled virtual environment.

Our main purpose consisted in designing an intelligent agent, the butler, for aiding disabled and elderly people in a virtual domotics environment. The agent must be adaptive, with the ability to react in real time, to learn, remembering and evolving with the experience, and to decide in specific situations in an autonomous way. Under these considerations, the users are allowed to interact with the assistive environment through voice, gesture and touch. In this sense, a large amount of data must be processed and inferred by the butler, which demands an adequate knowledge representation and efficient inferential mechanisms. In [6] some theoretical considerations about a knowledge representation model supporting complex relations between perceptions and actions are described. Also, a multiagent system [11] with several agents cooperating among them is required.

Related work can be found in [8] and [3], where a platform based on mobile agents combined with federated information management mechanism to create a flexible infrastructure for specialized care services, was developed. Attard et al. [2] illustrate a home system oriented ontology and an intelligent agent based framework for the rapid development of home control and automation. Velasco et al. [10] proposed an architecture for building a smart home environment using multiagent systems, demonstrating its effectiveness with an application example where multimedia contents follow the user movements throughout the house.

Section 2 describes the characteristics of the home environment; in section 3, the general aspects concerning the agent program are presented; finally, a practical implementation through a program prototype is discussed in section 4.

2 The Virtual Home Environment

Our task consisted of creating a virtual domotics house as the environment where the butler agent acts. The interaction with the house and the disabled user is through a set of domotics devices, in a reactive and continuous way, by receiving perceptions and taking the corresponding actions. Figure 1 shows a general scheme of such behaviour.

The environment progress is cyclic, with the agent receiving a set of perceptions –perception vector– coming from the domotics sensors of the house and the requests and/or the answers of the user. On the other hand, a daily planner provides the butler with the protocol related with the house and the disabled to be followed (user's lunch time and diet, medication, etc.). On the basis of the perception vector and the daily planner, the butler evaluates the house's and the user's states. As a result, a set of actions is planned –action vector–. The action vector contains three types of actions: actions on the environment (affect the disabled and the domotics house), actions on the daily planner (reprogram activities in it), and actions on the report of the disabled user (related to healthcare or diet considerations). Notice from figure 1 that actions on the environment may have direct consequences on the perception vector. At each time instant, the butler analyzes the perception vector and considers the perception with the highest priority, removing it from the perception. On the basis of the considered perception, the butler plans a set of actions to be performed; these actions will update the actual state of the environment.

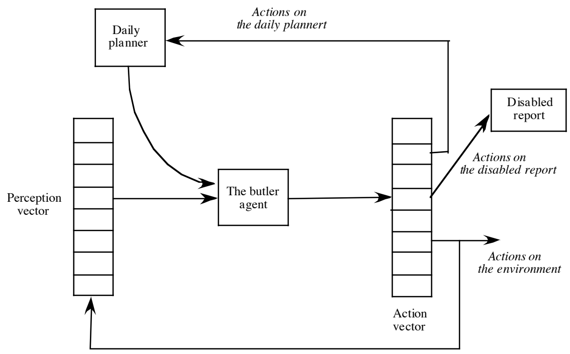

Fig. 1. Reactive behaviour of the butler

2.1 The Environment Specification

The environment specification considers the set of components and devices capable of generating perceptions as well as generating and/or receiving a set of actions. These actions are connected with the corresponding perceptions, in such a way that they describe the natural progress of the environment.

The basic components of the environment are: the domotics house, the user (disabled or elderly person), the daily planner, the user's report, and the butler agent. The domotics house, the disabled person and the daily planner are in charge of generating perceptions. Actions are taken by the butler, whose effects are manifested on the domotics house, the disabled person, the daily planner and the report of the user. The functional character of the environment is structured in five modules [7]:

1. The healthcare module, which functions are related with the health and medical prescriptions for user's treatments.

2. The nursing module, in charge of planning the daily menu of the user..

3. The preferences and scheduling module, by which the user's activities and hobbies preferences are scheduled and reprogrammed.

4. The stock module, that informs about the goods available and those exhausted in the house, such as food or medication.

5. The control devices module, which manages the control devices of the house. This module is made up of two components:

5.1. The house accommodation module, which covers the needs of accommodation of the user (light level, temperature, blinds, windows, etc.).

5.2. The house security module, in charge of covering the security matters of the user and the house (warning of electrical failures, water or gas leaks, fires, unexpected visitors, etc.).

2.2 The Environment Progress

The way that the environment evolves is determined by the connection between perceptions and actions. The perception-action vector (PAV) is the basic mechanism connecting perceptions with the actions to be taken in certain moment.

The PAV can be described as a set of *n* mappings, P_i: $V_i \rightarrow \Pi(A)$, i=1..n, P_i being the i-th perception, V_i is the domain of values of P_i, A is the set of possible actions on the environment, and $\Pi(A)$ is the set of parts of A. That is, the PAV associates each P_i value with the corresponding subset of actions to be taken. The PAV can be modeled through an array structure, such as figure 2 illustrates. Each set of actions associated with a given value of a perception will be considered independent of the moment it occurred.

P_1	$V^{1,1}$	a_i
		…
		a_j
	$V^{1,2}$	a_k
		…
		a_r
	…	…
	$V^{1,s}$	a_p
		…
		a_v
…	…	…
P_n	$V^{n,1}$	a_t
	…	…
	$V^{n,w}$	a_z

Fig. 2. The perception-action vector

It can be observed from figure 2 that $V_1=\{V^{1,1}, V^{1,2}, \ldots, V^{1,s}\}$, and value $V^{1,1}$ is associated with set $\{a_i, \ldots, a_j\}$ of actions, value $V^{1,2}$ with set $\{a_k, \ldots, a_r\}$.

The PAV provides a synthesized overview of the world progress. However, a wider range of general properties extending the capacity of the environment progress is also considered. Overcoming conflicting situations is a significant task. Conflicts appear when a goal can be reached through several possibilities, but only one of them can be considered. A particular case is when different subsets of actions are associated with the same perception –or subset of perceptions–.

Example: the Healthcare and the Control Devices Modules. In this prototype two modules have been considered: healthcare and control devices. Both modules contain a total of 36 perceptions coming from different components of the environment. Figure 3 shows a sketch of the environment progress in these situations, described in a PAV format. Perceptions in figure 3 are classified in several types, depending on their nature, and come from different sources of the environment. Each perception in figure 3 is identified by a code. Some of them are associated with more than one action, having a total of 39 different actions. Figure 4 illustrates the meaning of action symbols in figure 3.

Perceptions				
Type	Source	Code	Description	Actions
Generals	House	1	Date	
		2	Time	
		3	Position where the user stays in the current moment	
Physiological Parameters -PP-	Daily planner	4	It is time to take the PP to the user –temperature, hearth rhythm, blood pressure,…-	A1
				A2
	User	5	The user wants its PP to be taken	A2
	House	6	The health center notifies by e-mail that the PP of the user must be taken	A1
				A2
		7	Blood pressure	A3
				A4
		8	Hearth rhythm	A3
				A4
		9	Body temperature	A3
				A4
		10	Weight	A3
				A4
Take medication	Daily planner	11	It is time to take the medication	A5
New doctor's prescription	User	12	User informs the butler of a new doctor's prescription	A6
				A7
				A3
User don't feel well	User	13	User has a headache	A8
				A3
		14	User has a cold	A8
				A3
		…	…	…
Artificial illumination	User	20	The user asks for a change in artificial illumination	A9
		21	User asks butler for Turn light on / turn light off	A10
		22	User asks butler for more / less light intensity	A11
…	…	…	…	…
Emergency situation	House	36	House sensors detect an emergency situation	A38
				A39

Fig. 3. Sketch of the PAV corresponding to the healthcare and control devices modules

2.3 Information Systems as Decision Mechanisms

The agent's decision mechanism requires a full knowledge representation model of the environment, describing actions in terms of perceptions.

In the case of environments with simple perception-action relationship, a decision model based on a priority mechanism is enough to decide which perceptions are first considered and which actions are first performed. Nevertheless, when complex perception-action relations are established, a more powerful decision model is needed.

An information system (IS) is an abstraction model that constitutes a powerful tool to describe concepts in terms of properties or. An IS should also provide a mechanism to perform the corresponding decision.

In the context of this work, an IS provides a functional approach to describe actions in terms of perceptions. In particular, we deal with Object Attribute Tables (OAT), which can be understood as models of information systems.

An OAT is a mapping from a set NR={n_e_1, n_e_2,…, n_e_m} of n-tuples, to a set $\Pi(C)$ of subsets of concepts. C is a set of concepts and $\Pi(C)$ the set of parts of C; that is, OAT: NR $\rightarrow \Pi(C)$. Tuples n_e_i are made up of *n* values corresponding to *n* attributes of a set R={r_1, r_2,…, r_n}. Thus, n_e_i= ($v(r_1)$, $v(r_2)$,…, $v(r_n)$), $v(r_j) \in V_j$, j=1…n, $v(r_j)$ being the value of attribute $r_j \in$ R, whose domain is V_j.

If set R of attributes corresponds to a set P={p_1, p_2,…, p_n} of perceptions, and set C of concepts is identified by a set of A={a_1, a_2,…, a_w} of actions, then we are talking about a mapping connecting perceptions with actions. Formally, OAT:NP$\rightarrow \Pi$(A), NP being the set of n-tuples of values of the perception domains. Figure 5 shows a graphic illustration of the concept of OAT.

Symbol	Action meaning
A1	To notify the user about the imminent PP taking
A2	To take the PP to the user.
A3	To update the medical record
A4	If i t is not normal, then contact the health center.
A5	Take the medication to the user.
A6	Buy the new prescribed medication
A7	To update the daily planner
A8	Give the user an aspirin
A9	Change the state of artificial illumination.
A10	Turn light on / turn light off.
A11	Increase /decrease light intensity
…	…
A38	Evacuate the user
A39	Call the emergency services

Fig. 4. Meaning of some action symbols in figure 3

Sets {a_p, a_q,…, a_r} of figure 5 represent subsets of actions, that is, {a_p, a_q,…, a_r} $\subseteq$ A, and t_{ij} is the value of perception *j* corresponding to the i-th tuple. The meaning of the i-th row of an OAT is as follows: «if t_{i1} is the value of perception p_1 and t_{i2} the value of p_2 and…and t_{in} the value of p_n, then perform the subset {a_r, a_s,…, a_p} of actions».

NP \ P	p_1	p_2	⋯	p_j	⋯	p_n	A
n_e_1	t_{11}	t_{12}	⋯	t_{1j}	⋯	t_{1n}	{a_i, a_j,…, a_k}
n_e_2	t_{21}	t_{22}	⋯	t_{2j}	⋯	t_{2n}	{a_t, a_v}
	………………………………						
n_e_i	t_{i1}	t_{i2}	⋯	t_{ij}	⋯	t_{in}	{a_r, a_s,…, a_p}
	………………………………						
n_e_m	t_{m1}	t_{m2}	⋯	t_{mj}	⋯	t_{mn}	{a_x,…, a_z}

Fig. 5. Object Attribute Table

Object attribute tables have a higher descriptive power than perception-action vectors, since they allow expressing a subset of actions in terms of a set of perceptions. OAT also allows considering aspects closely related to complex environments, such as the existence of incomplete knowledge, the presence of a vast amount of perceptions, the capacity of handling quantitative and qualitative knowledge, and provides a way to extend the capabilities of agents towards tasks learning [4] [5]. Deciding what action must be performed in a given moment may require additional knowledge of the environment. This means that an extension of the model of the environment progress is required. Such extension is considered as an *internal state* of the environment [9]. The internal state may be useful for several purposes, for example, to adopt more precise actions or to avoid absurd decisions.

3. The Agent Program

The agent program is permanently monitoring the state of the disabled user, and it is based on a set of condition-action rules. At each step, the program takes the perception vector, the daily planner and the environment's state as the input data, and determines which OAT has to be considered. Based on the selected OAT, the agent decides the subset of actions to be performed.

Anyway, if two or more rules can be triggered in a given step –and so the corresponding object attribute tables must be considered–, only one of them will be selected. Such conflicting situations are overcome by associating a priority mechanism with rules. Rules with a higher priority are firstly evaluated. A given perception will only be taken into account when no perception with a higher priority is present in the perception vector. In this sense, perceptions to do with emergency situations have the highest priority, then those task referring to the health of the disabled are considered, and so on… Next, a simple sketch of a general agent algorithm is shown.

```
procedure BUTLER-AGENT(var:perception_vector; daily
planner; environment's state; var action_vector);
begin
     if P36 then performing the emergency_situation OAT;
     else
         if P13 OR P14 OR…  then performing the
                            user_don't_feels_well OAT;
         else
           …
end;
```

Symbols P_i of the above procedure refers to the i-th perception, according to figure 3.

A practical program prototype based on the discussed artificial intelligence approaches has been developed, showing the suitability and efficiency of the knowledge representation model adopted. Next section presents some aspects of the simulated environment, which the prototype is based on.

4. Simulation of a Home Domotics Environment

The main goal of this application is to simulate the interaction between a disabled person and his/her environment with the help of a virtual assistant.

The implemented modules of the system were the *control devices module* and the *healthcare module*. The *control devices module* takes into consideration various elements of the house: windows (*opening*/*closing*), curtains (*opening*/*closing*), lighting (*on*/*off*), and temperature (thermostat *on*/*off*). The *health care* module controls the state of health of the disabled person, monitors his physiological parameters and diagnosed diseases, and manages all the information related with medical prescriptions and medication times. The environment progress could be seen by providing the values for the parameters of each element of the control devices and healthcare modules –that is, the perceptions–, and creating simulated lists of tasks –the actions associated with the perceptions–. Thus, the behaviour of the intelligent agent could be evaluated.

Through a graphical interface, developed using C++ programming language and OpenGL, the user is able to change the status of each element. The interface of the application has been divided in 4 different windows. Figure 6 shows the graphical environment of the house with a disabled person in it.

Fig. 6. Corner view of the virtual house with the simulated disabled person

Figure 7(a) shows a window with the parameters of *time, environment, health parameters of the disabled person, the view of the camera and options*. At the moment, we are controlling the values of these parameters interactively, to see the results of the actions of the domotics elements and the butler as well. In the real system, the input to these parameters will come from the house itself and the disabled person. So, their simulation in this application is extremely important to refine the behaviour of the agent in extreme situations.

Figure 7(b) shows a window with the daily planner (*agenda*). The daily planner is implemented as a XML file, which contains the actions protocol to be followed by the butler. As the events of the daily planner occur they are shown in this window. Thus, we can evaluate the correspondence between the action of the butler and the action established in the daily planner.

Figure 7(c) shows the window corresponding to the *disabled record*. This file saves all the events that occur in the house as well as all the health related actions carried out by the butler. It is a very important component of the system, because it records all the events allowing us to detect any problem concerning the health of the disabled, so that it can be corrected in the real system.

Another important aspect of the simulation was the introduction of emotions in the butler behaviour. The considered emotions were the basic ones: *joy*, *sadness*, *anger*, *disgust*, *surprise*, and *fear*. The reason to do this was the need for an expressive and realistic interface in order to get interactions as natural as possible between the butler and the disabled person.

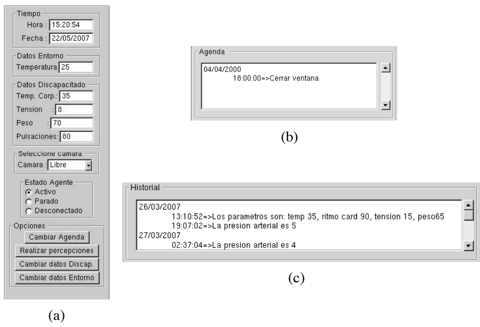

Fig. 7. (a) Interface controls for *time, environment data, disabled information, camera and options*; (b) Interface of the daily planner –*agenda*–; (c) Interface of the *disabled record*

Each emotion was associated with some events, which determine the conditions under which the emotion is triggered. Thus, an event like a noticeable elevation of the blood pressure of the patient elicits the emotion of *sadness* if we have a sensitive agent, or *surprise* if the agent did not expect it. The process to evaluate the recognition of emotions in facial expressions of synthetic faces is found in [1]

5. Conclusions

Some major issues affecting the design of an intelligent agent for home assistance have been discussed. First, the environment specification made up of five functional modules was described. Next, a first level specification of the way that the environment evolves was discussed. The underlying model of such specification is known as the perception-action vector (PAV). Then, the basic mechanisms supporting the decision-making process of the agent, known as information system (IS), were described.

Finally, a prototype simulating a home domotics environment has been presented. It provided a tool to visualize how the events of the world would be triggered and how the actions of the agent would be performed. Interactivity was achieved by a set of control interfaces, which allowed the user to change the parameters of the environment, of the health of the patient, and the daily planner.

Thus, we could prove that situations are correctly handled by the underlying logic of the butler agent. The visualization of the emotional state of the butler was also a very important issue, because human-computer interaction needs a believable and realistic character, which is accomplished using emotional states.

In the final system, the environment itself will provide input parameters for the butler, and a complete virtual avatar capable of moving and talking will be responsible for the interaction between the disabled person and the agent.

Acknowledgments. This work has been partially supported by the Direccion General de Investigación del Ministerio de Educación, Ciencia y Tecnología, through the TIN2007-67993 and the TIN2007-63025 research projects. We are also grateful to Dr. M. Miró, Mr. G. Trias, Ms. C. Blanco, Mr. D. García, Mr. S. Lora, Mr. E. Sigg, Mr. J. Simó and Mr. A. Sobrino, for their collaboration.

References

1. Arellano D, Lera I, Varona J, Perales, F J (2008) Virtual Characters with Emotional States. To be presented in 2nd PEACH Summer School
2. Attard M, Montebello M (2006) DoNet: a semantic domotic framework. In: Proceedings of the 15th international Conference on World Wide Web WWW '06. ACM, New York, NY, 997-998
3. Camarinha-Matos L M, Castolo O, Rosas, J (2003) A multi-agent based platform for virtual communities in elderly care. In: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, ETFA '03, vol.2, 421-428
4. Fiol G (1999) UIB-IK: A Computer System for Decision Trees Induction. In: Ras, Z.W., Skowron, A., (eds. LNCS, 1609, 601--611. Springer, Heidelberg
5. Fiol G (2003) Inductive Learning from Incompletely Specified Examples. Frontiers in Artificial Intelligence and Applications. 100: 286--295
6. Fiol G (2007) Intelligent Agent for Home Assistance. In: 11th IASTED International Conference on Artificial Intelligence and Soft Computing, 102--107. ACTA Press, Calgary
7. Muñoz C, Arellano D, Perales F J, Fontaned G (2006) Perceptual and Intelligent Domotic System for Disabled People. In: Proceedings of the 6th IASTED International Conference on Visualization, Imaging and Image Processing,70--75. ACTA Press, Calgary
8. Perales F J, Fiol G, Varona X, Miró M, Fuster P, Cerezo E, Baldassarri S, Remiro V, Serón F J, Pina A, Azkue I (2005) El Proyecto INEVAI3D: Agentes Autónomos 3D, Escenarios Virtuales e Interfaces Inteligentes para Aplicaciones de Domótica y de R-ealidad Virtual. In: 1th CEDI Spanish Conference on Informatics, 479--486. Thomson Editores Spain, Paraninfo S.A.
9. Russell S J, Norvig P (2004) Artificial Intelligence: A Modern Approach (2nd Ed.). Prentice Hall
10. Velasco J R, Marsá-Maestre I, Navarro A, López M A, Vicente A J, de la Hoz E, Paricio A, Machuca M (2005) Location-aware services and interfaces in smart homes using multiagent systems. In: The 2005 International Conference on Pervasive Systems and Computing
11. Wooldridge M (2002) An Introduction to Multiagent Systems. John Wiley&Sons