

A Persuasive Cognitive Assistant System

Angelo Costa¹, Stella Heras², Javier Palanca², Paulo Novais¹, Vicente Julián²

¹ ALGORITMI Research Centre, University of Minho, Portugal
{acosta, pjon@di.uminho.pt}

² Departamento de Sistemas Informáticos y Computación, Universitat Politècnica de València, España {sheras, jpalanca, vinglada@dsic.upv.es}

Abstract. In this paper, we present a persuasive recommendation module to be included into the iGenda framework. iGenda is a cognitive assistant that helps care-receivers and caregivers in the management of their agendas. The proposed new module will allow the system to select and recommend to the users the action that potentially best suits to his/her interests. The multi-agent approach followed by the iGenda framework facilitates an easy integration of these new features.

Keywords: iGenda, AAL, Persuasive System, Persuasion, Recommendation

1 Introduction

Ambient Assisted Living (AAL) applications possess the capability to promote security and comfort for the users, trying to provide an integrated solution that connects several distinct devices to services to form a unique solution [1]. Current AAL applications can perceive the environment, monitoring events, and provide an adjusted and timely response that enables it to interact with the users. However, these features are not enough by themselves. The aim is to accommodate people, and people change their needs and behaviors. These important aspects are not taken into account in current AAL applications. Specifically, current AAL applications suffer from different problems to be directly used to persuade human behavior, mainly related with the social environment of each individual. Often the target audiences are large and heterogeneous, and include users with wide-ranging goals, needs, and preferences. Thus, persuading the entire audience effectively with a one-size-fits-all persuasive intervention is difficult [2].

On the other hand, over the last few years, technology has been evolving in pursuit of persuading users and motivating them toward specific individual/collective behaviors. One of the most employed technology in order to try to persuade users is recommender systems. Current limitations of these systems are that traditional recommender systems base their recommendations on quantitative measures of similarity between the user's preferences and the current items to recommend (i.e. content-based recommenders [3]), between the user's profile and the profile of other users with similar preferences (i.e. collaborative filtering recommenders [4]) and on combinations of both (i.e. hybrid recommenders [5]). However, [6] has stated the inability of current recommender systems to use the large amount of qualitative and quantitative data available in AAL domains to

empower recommendations. Usually, recommender systems do not provide an explanation about the reasoning process that has been followed to come up with specific recommendations. However, this does not follow current trends, where people trust recommendations more when the engine can provide reasons for them [7]. Thus, what is understood as a good recommendation is changing from the one that minimizes some error evaluation to the one that really makes people happier. Several studies demonstrated that the argumentation of the different recommendations has a strong impact to strengthen these recommendations, including direct feedback from the user and providing alternative options which better fits the user needs and expectations [6][8]. The personalized selection of arguments can streamline the persuasion task and make the desired behavior or attitude change easier to achieve.

In order to overcome these problems, our proposal is to embed a new persuasive layer into the iGenda platform [9, 10], taking into account aspects such as the generation of arguments that support tasks recommended by the AAL application to the users. Our work involves a contribution in the above commented areas, presenting a persuasive recommendation module to be used in iGenda in order to improve its system credibility.

2 The iGenda framework

The iGenda is a cognitive assistant built upon a multi-agent system. It's purpose is to help care-receivers and caregivers by managing their agendas (create and reallocate events) and by presenting an easy-to-use visual interface for web and mobile usage, the platform is presented extensively in the papers [9, 1]. The aim of the iGenda is to be integrated on the users daily life and to become a constant helper, reminding the users of their appointments and events. It promotes active aging by introducing playful events on the users free time. Moreover, it serves as an visual interface to the users, showing new events and notifications and allowing the creation of new events. It is designed to be used by the caregivers and the care-receivers alike, having each one a specially designed interface that accounts for their roles on the platform.

One of the issues with iGenda, and most of the AAL platforms, is that it does not gather the users input or opinion. Works by Holzinger et al. [11] and Lindley et al. [12] have showed that it is imperative that the users feel included and part of the decision process. It is hard to release the control of decision making after a life of making choices. In iGenda when an event is scheduled it is showed to the user but not explained why or give any option to decline or accept the event. The lack of reason and justification of why the events should be performed may lead to the withdrawal of the system by the users. A major advance would be to include persuasive methods that may provide a motive to that specific event scheduling and compel the users to attend to it.

In terms of the Free Time Manager (which promotes active aging through the scheduling of activities on the care-receivers free time) most of the activities scheduled by it are optional, but they greatly help the users to be active and

promote an healthy life, but without a justification it is hard to put them in context or why they matter. Each time iGenda detects that a new action must be proposed to deal with the needs of the user, it selects a set of matching actions from its *free time events database*. In the previous version of iGenda, if the system was able to find more than one matching action to offer to the user, a single event was selected trough a biased random function; the selection leans towards the best ranked six events or an unbiased random event, the choice between the two functions is also random. However, the potential willingness of the user to accept a specific action (based on his/her current social context - i.e. the specific user, the specific caregiver, their relation, etc. - and the knowledge of similar past experiences) was not taken into account.

One of the main features is the modular architecture that the iGenda has. Its open connectivity allows the connection of new modules that follow the protocols [13, 14]. Being a multi-agent system, the iGenda allows easy integration of new features and the deployment and removal of agents at once [15].

3 Persuasion Module

In this section, we present the new persuasion module that we have added to the iGenda tool. The new persuasion module enhances the performance of iGenda, by allowing the system to select and recommend to the user the action that potentially best suits to his/her social context and interests. In addition, the user is presented with reasons that support the suitability of the action proposed. Thus, in this way the iGenda system tries to persuade the user to accept the action and to motivate his/her to put it into practice.

The persuasive module is based on the case-based argumentation framework for agent societies presented in [16]. Therefore, when iGenda has to select a specific event among a set, the system tries to create one argument (or more) to support each action. Then, an internal argumentation process takes part to decide the action that is better supported by its arguments.

3.1 Argumentation Framework

In this work, we have applied Agent-specific Argumentation Framework in an Agent Society (*AAFAS*) presented in [16] to determine which agent's argument attacks another agent's argument in an argumentation process performed in a society of agents and, in each case, which argument would defeat the other. To do that, we have to consider the values that arguments promote (the preferences of the users), the users' preference relations (preference orderings), and the dependency relations between agents (the relations that emerge from agent interactions or are predefined by the system).

Hence, our system models a society of agents with a set Ag of agents of the agent society S , a set of Rl agents' roles that have been defined in S , a set of D possible dependency relations in S , and a set of V values predefined in S . Thus, an agent specific argumentation framework for an agent society is a tuple $AAFAS = \langle Ag, Rl, D, V, A, Role, Dependency_S, val, Valpref_{agi} \rangle$ where:

Definition 1 (Agent-specific AF for an Agent Society).

- $Ag, Rl, D,$ and V are defined as the above elements of the agent society.
- A is the set of arguments of the argumentation framework.
- $Role(ag, a) : Ag \times A \rightarrow Rl$ is a function that assigns an agent the specific role that it plays (from its set of roles) when it has put forward a specific argument.
- $Dependency_S : <_D^S \subseteq Rl \times Rl$ defines a reflexive, transitive and asymmetric partial order relation over roles.
- $val(ag, a) : Ag \times A \rightarrow 2^V$ is a function that assigns an agent's argument the value(s) that it promotes.
- $Valpref_{ag_i} \subseteq V \times V,$ defines an irreflexive, transitive and asymmetric relation $<_{ag_i}^S$ over the agent's ag_i values in the society S .

In our system, agents can play the role of *patients*, *caregivers* (which can be relatives, personal health assistants, friends, etc.), and *doctors*. We also consider the following dependency relations: (i) *Power*: when an agent has to accept a request from another agent because of some pre-defined domination relationship between them. For instance, in our agent society S , $Patient <_{Pow}^S Doctor$, and $Caregiver <_{Pow}^S Doctor$ since patients and caregivers must follow the guidelines recommended by their doctors; (ii) *Authorisation*: when an agent has committed itself to another agent for a certain service and a request from the latter leads to an obligation when the conditions are met. For instance, in S , $Patient <_{Auth}^S Caregiver$, if the patient has contracted the health assistant service that a caregiver offers; and (iii) *Charity*: when an agent is willing to accept a request from another agent without being obliged to do so. For instance, in S , by default $Patient <_{Ch}^S Patient$, $Caregiver <_{Ch}^S Caregiver$ and $Doctor <_{Ch}^S Doctor$.

In addition, values represent preferences about the different types of activities that iGenda can recommend to the user. Concretely, in our system we have established the following typology of values and activities:

- *Motion Values*: which represent preferences for activities that are performed still or sitting (*motionless*), standing up or with little movement (*low-motion*), or those performed with physical effort (*motion*).
- *Location Values*: which represent preferences for activities that are performed indoors without movement (*homebound*), indoors with movement (*indoors*), outdoors with movement (*outdoors*), or outdoors with movement but that implies that the person has to be carried (*limited-outdoors*).
- *Social Values*: which represent preferences for activities that involve socialise with others (*social*), or not (*individual*).
- *Environmental Conditions Values*: which represent preferences for activities that only can be performed with good weather (*weather-dependent*), or not (*weather-independent*).
- *Health Conditions Values*: which represent preferences for activities that have immediate or direct impact on health (*mandatory*), or not (*not-mandatory*).

Table 1: Structure of an Argument Case AC1

PROBLEM	Domain Context	Premises = {Sensors information, etc.}	
	Social Context	Proponent	ID = Doctor1 (D1)
			Role = Doctor
			$ValPref_{D1} = [\text{indoors} < \text{outdoors}]$
		Opponent	ID = Caregiver1 (C1)
			Role = Patient
Dependency Relation = Power			
SOLUTION	Conclusion = $S1$ (Outdoors gymnastics activity)		
	Acceptability State = Unacceptable		
	Received Attacks	Distinguish Premises = $Rain$	
		Counter Examples = {AC2}	
JUSTIFICATION	Guidelines = {G.1.2}		

We have adapted one of the knowledge resources of this framework, the *argument-cases case base*, to use it as a persuasion resource for our system. This resource stores previous experiences and their final outcome in the form of case-based arguments, which can be retrieved and used later: 1) to generate new arguments that support each action; 2) to select the best action to recommend in view of past experiences; and 3) to store the new argumentation knowledge gained in each process, improving the system persuasion skills. The argument-cases are the main structure that we use to implement our framework and computationally represent arguments in agent societies. Table 1 shows an example of the structure of a specific argument-case in our system. The argument-cases have three main parts: the description of the *problem* that the case represents (i.e. the features that describe the situation where an action has to be recommended), the *solution* applied to this problem (i.e. the action recommended) and the *justification* why this particular solution was applied (i.e. specific features that match the situation, guidelines that have motivated the recommendation of a specific action, etc.). Therefore, an argument-case stores the information about a previous argument that an agent created to support the recommendation of a specific action. **Problem:** The problem description stores the *premises* of the argument-case, which represent the context of the domain where the case was created. In addition, if we want to store an argument and use it to generate a persuasive argument in the future, the features that characterise the audience of the previous argument (the social context) must also be kept.

Therefore, we store in the argument-case the social information about the *proponent* of the argument, the *opponent* to which the argument is addressed, and the dependency relation established between the roles that these agents play. Thus, the proponent and opponent’s features represent information about the agent that generated the argument and the agent that received it respectively. Concretely, for each agent the argument-case stores a unique *ID* that identifies it in the system and the *role* that the agent was playing when the argument was created. Moreover, if known, we also store the preferences of each agent over

the pre-defined set of general values in the system. These preferences affect the persuasive power of the proponent's argument over the opponent's behaviour. Finally, the dependency relation between the proponent's and the opponent's roles is also stored.

Solution: In the solution part, the *conclusion* of the argument (the action that the argument-case supports) is stored. Moreover, the argument-case stores the information about the *acceptability state* of the argument at the end of the dialogue. This feature shows if the argument was deemed *acceptable*, *unacceptable* or *undecided* in view of the other arguments that were put forward. Regardless of the final acceptability state of the argument, the argument-case also stores the information about the possible *attacks* that the argument received. These attacks could represent the justification for an argument to be deemed unacceptable or else reinforce the persuasive power of an argument that, despite being attacked, was finally accepted. Argument-cases can store different types of attacks, depending on the type of argument that they represent: premises which value in the context where the argument was posed was different (or non-existent) than the value that it took when the argument-case was generated (*distinguish the case*) or argument-cases which premises also match the premises of the context where the argument was posed, but which conclusion is different than the conclusion of the case(s) used to generate the argument (*counter-examples*).

Justification: The justification part of the argument-case stores the information about the knowledge resources that were used to generate the argument represented by the argument-case. In our system, the justification stores the information of the health guidelines used to recommend a specific action for a specific user in a particular situation.

Following a CBR methodology, the knowledge resources of the agents' case-based argumentation system allow them to automatically generate, select and evaluate arguments. However, the complete argument management process (how agents generate, select and evaluate arguments by using the knowledge resources of their argumentation systems) is out of the scope of this paper. Also, the framework presented is flexible enough to represent different types of arguments and their associated information, but the value of some features on argument-cases could remain unspecified in specific domains. For instance, in some open MAS, the preferences over values of other agents could not be previously known. However, agents could try to infer the unknown features by using CBR adaptation techniques. Therefore, in our proposal, arguments that iGenda uses to are tuples of the form:

Definition 2 (Argument). $Arg = \{\phi, p, \langle SS \rangle\}$, where ϕ is the conclusion of the argument, p is the preference value that the argument promotes and $\langle S \rangle$ is a set of elements that justify the argument (the support set).

The support set $\langle SS \rangle$ is the set of features (*premises*) that represent the context of the domain where the argument has been put forward (those premises that match the problem to solve and other extra premises that do not appear in the description of this problem but that have been also considered to draw the conclusion of the argument) and optionally, any knowledge resource used by the

proponent to generate the argument (e.g. the health guidelines). On the other hand, the support set can also include any of the allowed attack elements of our framework. These are: *distinguishing premises*, or *counter-examples*.

Now, the concept of conflict between arguments defines in which way arguments can attack each other. There are two typical attacks studied in argumentation: *rebut* and *undercut*. In an abstract definition, rebuttals occur when two arguments have contradictory conclusions (i.e. if an argument a_1 supports a different conclusion for a problem description that includes the problem description of an argument a_2). Similarly, an argument undercuts other argument if its conclusion is inconsistent with one of the elements of the support set of the latter argument or its associated conclusion (i.e. if the conclusion drawn from the argument a_1 makes one of the elements of the support set of the argument a_2 or its conclusion non-applicable in the current recommendation situation). Thus, we can define the agent-specific defeat relation of *AAFAS* as:

Definition 3 (Defeat). *An agent's ag_1 argument $a_1 \in AAFAS$ that is put forward in the context of a society S defeats $_{ag_1}$ another agent's $ag_2 \in AAFAS$ argument a_2 iff $attack(a_1, a_2) \wedge (val(ag_1, a_1) <_{ag_1}^S val(ag_1, a_2) \notin Valpref_{ag_1}) \wedge (Role(ag_1) <_{Pow}^S Role(ag_2) \vee Role(ag_1) <_{Auth}^S Role(ag_2) \notin Dependency_S)$*

3.2 Example

To exemplify our framework, let us propose a simple scenario of an open MAS that represents a situation where iGenda has to schedule a health-care activity for a patient $P1$. Thus, iGenda retrieves from its activities database a potential activity $S1$ that was proposed by the patient's doctor $D1$ by following a health guideline G.1.2 and recommends an outdoors gymnastics activity that promotes the value *outdoors*. Also, it retrieves another activity $S2$ that was proposed by the patient's caregiver $C1$ by following another health guidelines and recommends an indoors training that promotes the value *indoors*. The doctor $D1$ has a value preference order $ValPref_{D1} = [indoors < outdoors]$ that promotes outdoors activities over indoors. On the contrary, let us assume that both the patient and his caregiver $C1$ have a value preference order $ValPref_{C1} = [outdoors < indoors]$ that represent their preference for indoors activities over outdoors. In addition, as established by our system, doctors have a power dependency relation over caregivers and patients that forces patients and caregivers to accept the recommendations provided by doctors. In this context, the iGenda system would internally generate a support argument (and its associated argument-case) for each possible activity to recommend as follows:

$A1 = \{S1, outdoors, < Patient1, SensorsInformation, G.1.2 >\}$ In view of the current context, Doctor $D1$ proposes the outdoors activity $S1$ following the health guidelines G.1.2. $A2 = \{S2, indoors, < Patient1, SensorsInformation, G.3.2 >\}$ In view of the current context, Caregiver $C1$ proposes the indoors activity $S2$ following the health guidelines G.3.2, which promotes the value indoors that matches the preference of the patient to do indoors activities over outdoors.

Up to this point, iGenda would have to schedule the activity $S1$ for the patient, since the *power* dependency relation of doctors over caregivers and patients will prevail, $A1$ *rebutts* $A2$, and $A2$ would be defeated by $A1$. However, let us assume that the day is rainy. With this new information, the system would be able to find and retrieve the argument-case $AC2$, which stores the information about a similar situation where an indoors activity $S2$ was proposed by the doctor due to the bad weather conditions. Then, the system will generate the following counter-example for the argument $A1$:

$A2.1 = \{S2, indoors, \langle Patient1, Rain, AC2 \rangle\}$ In view of the current context, Caregiver $C1$ proposes the indoors activity $S2$ following the health guidelines G.3.2, which promotes the value indoors, matches the preference of the patient to do indoors activities over outdoors, and takes into account the current rainy weather conditions.

Clearly, $A2.1$ *undercuts* $A1$, since $A1$ did not take into account the new premise *rain* and hence, $A1$ does not longer match the current situation of the patient. Therefore, $A1$ would be defeated and iGenda would recommend the activity $S2$ for the patient. If no more activities and their associated support arguments are posed, iGenda will finish the argumentation process and will store the new argument-case $AC1$ (as represented in 1). This new argument-case improves the recommendation skills of the system by representing the knowledge gained in the above argumentation process. Then, from now on and unless other argumentation process updates the information of the argument-cases case-base, the outdoor activity $S1$ would never be proposed again in rainy weather conditions (although it matches the preferences of the doctor). In addition, the argument-cases can be used to prompt persuasion messages for the user of iGenda. For instance, the system may create persuasion messages as “iGenda recommends you to perform activity $S1$ since you prefer indoor activities”, “iGenda recommends you to perform activity $S1$ since is rainy” or “iGenda recommends you to perform activity $S1$ since it follows the guidelines of your doctor when rains”.

4 Conclusions

This paper has presented a new persuasive oriented module to be included into the iGenda framework. The persuasive module is a case-based argumentation approach which allows iGenda not only to select a specific event among a set, but also it allows the system to create one argument (or more) to support each possible event. To do this, an argumentation process takes part into the system to decide the action that is better supported by its arguments. This process enhances the response given by iGenda to the user because the selected action is presented with reasons that supports it. Thus, the proposed solution tries to persuade the user to accept the action and to put it into practice.

Acknowledgements

This work has been partially supported by the projects TIN2015-65515-C4-1-R and TIN2014-55206-R of the Spanish government, by the grant program for the recruitment

of doctors for the Spanish system of science and technology (PAID-10-14) of the Universitat Politècnica de València, and by COMPETE: POCI-01-0145-FEDER-007043 and FCT – Fundação para a Ciência e Tecnologia within the projects UID/CEC/00319/2013 and Post-Doc scholarship SFRH/BPD/102696/2014 (A. Costa).

References

1. Costa, Â., Castillo, J.C., Novais, P., Fernández-Caballero, A., Simoes, R.: Sensor-driven agenda for intelligent home care of the elderly. *Expert Systems with Applications* 39 (2012) 12192–12204
2. Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. *User Modeling and User-Adapted Interaction* 18 (2007) 245–286
3. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: *The Adaptive Web*. Springer (2007) 325–341
4. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: *The Adaptive Web*. Springer (2007) 291–324
5. Burke, R.: Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12 (2002) 331–370
6. Chesñevar, C., Maguitman, A.G., González, M.P.: Empowering recommendation technologies through argumentation. In: *Argumentation in Artificial Intelligence*. Springer (2009) 403–422
7. Linden, G., Hong, J., Stonebraker, M., Guzdial, M.: Recommendation algorithms, online privacy, and more. *Communications of the ACM* 52 (2009) 10
8. Heras, S., Navarro, M., Botti, V., Julián, V.: Applying dialogue games to manage recommendation in social networks. In: *Lecture Notes in Computer Science*. Springer (2010) 256–272
9. Costa, A., Novais, P., Corchado, J.M., Neves, J.: Increased performance and better patient attendance in an hospital with the use of smart agendas. *Logic Journal of IGPL* 20 (2011) 689–698
10. Costa, A., Novais, P., Simoes, R.: A caregiver support platform within the scope of an ambient assisted living ecosystem. *Sensors* 14 (2014) 5654–5676
11. Holzinger, A., Ziefle, M., Röcker, C.: Human-computer interaction and usability engineering for elderly (HCI4Aging): Introduction to the special thematic session. In: *Lecture Notes in Computer Science*. Springer (2010) 556–559
12. Lindley, S., Wallace, J.: Placing in age. *ACM Transactions on Computer-Human Interaction* 22 (2015) 1–39
13. Andrade, F., Novais, P., Machado, J., Neves, J.: Contracting agents: legal personality and representation. *Artificial Intelligence and Law* 15 (2007) 357–373
14. Andrade, F., Neves, J., Novais, P., Machado, J., Abelha, A.: Legal security and credibility in agent based virtual enterprises. In: *IFIP — The International Federation for Information Processing*. Springer (2005) 503–512
15. Castillo, J.C., Carneiro, D., Serrano-Cuerda, J., Novais, P., Fernández-Caballero, A., Neves, J.: A multi-modal approach for activity classification and fall detection. *International Journal of Systems Science* 45 (2013) 810–824
16. Heras, S., Botti, V., Julián, V.: Argument-based agreements in agent societies. *Neurocomputing* 75 (2012) 156–162