



HAL
open science

Using Argumentation for Ambient Assisted Living

Julien Marcais, Nikolaos Spanoudakis, Pavlos Moraitis

► **To cite this version:**

Julien Marcais, Nikolaos Spanoudakis, Pavlos Moraitis. Using Argumentation for Ambient Assisted Living. 12th Engineering Applications of Neural Networks (EANN 2011) and 7th Artificial Intelligence Applications and Innovations (AIAI), Sep 2011, Corfu, Greece. pp.410-419, 10.1007/978-3-642-23960-1_48 . hal-01571468

HAL Id: hal-01571468

<https://inria.hal.science/hal-01571468v1>

Submitted on 2 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Using Argumentation for Ambient Assisted Living

Julien Marcais¹, Nikolaos Spanoudakis² and Pavlos Moraitis¹,

¹ Laboratory of Informatics Paris Descartes (LIPADE) – Paris Descartes University, France
Julien.marcais@gmail.com, pavlos@mi.parisdescartes.fr

² Department of Sciences - Technical University of Crete, Greece
nikos@science.tuc.gr

Abstract. This paper aims to discuss engineering aspects for an agent-based ambient assisted living system for the home environment using argumentation for decision making. The special requirements of our system are to provide a platform with cost-effective specialized assisted living services for the elderly people having cognitive problems, which will significantly improve the quality of their home life, extend its duration and at the same time reinforce social networking. The proposed architecture is based on an agent platform with personal assistant agents that can service users with more than one type of health problems.

Keywords: Multi-Agent Systems, Ambient Assisted Living, Argumentation, Alzheimer disease.

1 Introduction

This paper is concerned with addressing the non-trivial task [9] of engineering an AAL information system addressing the needs of the elderly suffering from dementia (Alzheimer Disease) with identified risk factors as well as having cognitive problems (referred to as *end users* from now on). This application domain is about offering indoor assistance aiming to enhance the autonomy and quality of life of the user.

Agent-based computing is considered as one of the most appropriate technologies for ambient assisted living systems in an interesting survey presented by Becker [1]. Another interesting survey on agent-based intelligent decision support systems to support clinical management and research can be found in the work of Foster et al. [3]. Finally interesting information about the characteristics of residential monitoring applications designed to be used in consumers' personal living spaces and more particularly to be used by persons with dementia is presented by Mahoney et al. [7].

The HERA project¹ aims to build an AAL system to provide cost-effective specialized assisted living services for the elderly people suffering from MCI or mild/moderate AD or other diseases (diabetes, cardiovascular) with identified risk

¹ The “Home sERVICES for specialised elderly Assisted living” (HERA, <http://aal-hera.eu>) project is funded by the Ambient Assisted Living Joint Programme (AAL, <http://www.aal-europe.eu>)

factors, which will significantly improve the quality of their home life, extend its duration and at the same time reinforce social networking. In order to achieve this goal, HERA will provide to these end users the following main categories of services:

- Cognitive Exercises, the end users play cognitive reinforcement games at their TV using the remote control
- Passive communication, the end user can select to watch informative videos and news items related to his/her disease
- Pill and Exercise reminders, the end user is reminded for taking his pills or doing his exercises while he is watching TV or a video, or is playing a game
- Reality orientation, date and time is visible on screen while watching videos or the TV, also while doing their exercises
- Blood pressure and weight monitoring, the user can perform the measurements at specific intervals and he can monitor his condition along with his doctor

HERA applies technological solutions for aiding users managing their daily lives. Thus, by using the HERA system, the time to be at home, rather than in an institution, will be prolonged and relieve them from visiting the specialists often, while keeping them able to perform their daily activities and social interactions.

In [13] we presented the HERA system requirements and proposed an architecture to address the challenges of engineering such systems (see, e.g., Kleinberger et al. [6]):

- *Adaptability*. No two human beings have the same needs or everyday life habits. An AAL system must be able to adapt to a particular user. For the agents decision making we chose argumentation as it allows for decision making using conflicting knowledge, thus different experts can express their opinion that can be conflicting.
- *Natural and anticipatory Human-Computer Interaction (HCI)*. The people that need assistance very often have limitations and handicaps. In HERA the use of the TV set and remote control is ensuring a quick learning curve for our users,
- *Heterogeneity*. AAL systems are expected to be capable of being integrated with several subsystems developed by different manufacturers (e.g. sensors, etc). We use a service oriented architecture based on web services that allows the different sub-systems to be connected in a plug and play standardized way.

There are specific advantages of our approach compared with previous work (see e.g. Bravo et al. [2], or García et al. [4]). Firstly, the autonomy of the user is increased and the ambient assistance is automated. Secondly, the use of argumentation allows for decision making even in cases when a user has more than one chronic diseases situations.

The system service oriented architecture and evaluation process is presented in [11]. In this paper we focus in presenting the use of argumentation in HERA by one of the most important modules the multi-agent system.

In what follows we briefly discuss argumentation technology, then we present an overview of the agents design process and, finally, we focus in the knowledge representation and reasoning in HERA.

2 Argumentation Technology

Argumentation can be abstractly defined as the formal interaction of different conflicting arguments for and against some conclusion due to different reasons and provides the appropriate semantics for resolving such conflicts. Thus, it is very well suited for implementing decision making mechanisms dealing with the above requirements. Moreover, the dynamic nature of those conflicting decisions due to different situations or contexts needs a specific type of argumentation frameworks (such as those proposed by Kakas and Moraitis [5] or Prakken and Sartor [10]). These frameworks are based on object level arguments representing the decision policies and then they are using priority arguments expressing preferences on the object level arguments in order to resolve possible conflicts. Subsequently, additional priority arguments can be used in order to resolve potential conflicts between priority arguments of the previous level. Therefore, we are concerned with argumentation frameworks that allow for the representation of dynamic preferences under the form of dynamic priorities over arguments. In this work we are using the framework proposed by Kakas and Moraitis [5]. This framework has been applied in a successful way in different applications (see e.g. [8], [14]) involving similar scenarios of decision making and it is supported by an open source software called Gorgias². The latter allows for defining dynamic priorities between arguments, which means that the priorities of rules can depend on context. Finally, the modularity of its representation allows for the easy incorporation of views of different experts.

The Gorgias Framework

Gorgias is an open source implementation of the framework presented above in the Prolog language. Gorgias defines a specific language for the representation of the object level rules and the priorities rules of the second and third levels. The language for representing the theories is given by rules with the syntax in formula (1).

$$\text{rule}(\text{Signature}, \text{Head}, \text{Body}). \quad (1)$$

In the rule presented in formula (1), Head is a literal, Body is a list of literals and Signature is a compound term composed of the rule name with selected variables from the Head and Body of the rule. The predicate prefer/2 is used to capture the higher priority relation (h_p) defined in the theoretical framework. It should only be used as the head of a rule. Using the syntax defined in (1) we can write the rule presented in formula (2).

$$\text{rule}(\text{Signature}, \text{prefer}(\text{Sig1}, \text{Sig2}), \text{Body}). \quad (2)$$

Formula (2) means that the rule with signature Sig1 has higher priority than the rule with signature Sig2, provided that the preconditions in the Body hold. If the modeler needs to express that two predicates are conflicting he can express that by using the rule presented in formula (3).

² Gorgias is an open source general argumentation framework that combines the ideas of preference reasoning and abduction (<http://www.cs.ucy.ac.cy/~nkd/gorgias>)

$$\text{conflict}(\text{Sig1}, \text{Sig2}). \quad (3)$$

The rule in formula (3) indicates that the rules with signatures Sig1 and Sig2 are conflicting. A literal's negation is considered by default as conflicting with the literal itself. A negative literal is a term of the form neg(L). There is also the possibility to define conflicting predicates that are used as heads of rules using the rule presented in formula (4).

$$\text{complement}(\text{Head1}, \text{Head2}). \quad (4)$$

Development Environment

The development environment involves several tools:

- Gorgias
- Logtalk
- SWI-prolog

In Prolog we use a Knowledge Base (KB), which is composed of Horn clauses. There are two types of clauses, facts and rules. Rules follow this form:

$$\text{Head} \text{ :- } \text{Body}. \quad (5)$$

This means that the head of this rule will be true if and only if all the predicates which compose the body are true. The head is a single predicate. Facts are rules without body.

The developer has just to describe the world with facts and rules and then ask a question to Prolog. There are many Prolog implementations, one of which is SWI-Prolog (the free prolog environment that we used).

Logtalk is the bridge between two worlds: Logic Programming and Object-Oriented Programming. It is developed as a Prolog extension that's why Logtalk needs a Prolog compiler. A program written with the Logtalk language will first be translated by the Logtalk compiler into a full Prolog file. Then the generated Prolog file is compiled by the selected Prolog compiler. It is needed in order to be able to run simultaneously more than one prolog instances using the Java JPL interface.

3 HERA Multi-Agent System Analysis

The heart of the HERA services platform is the Multi-Agent System (MAS). For designing our agents we used the ASEME methodology [12]. In Figure 1 the System Actors Goals (SAG) model is presented. Actors are depicted as circles filled with black color, while their goals are depicted in rounded rectangles. An actor's goal can be dependent to another actor, in this case there are directed lines from the dependor to the goal and then from the goal to the dependee.

A personal assistant (PA) agent serves a specific user and has access to the user's profile data. This agent is proactive in the sense that he is always active following the user's agenda and providing support and advice by taking action whenever needed. For example he is able to send a message to the user for reminding him to take his pills, to receive sensor information about the user and consult experts' knowledge on

the user’s behalf. Moreover, the PA uses the requests’ history in order to adapt the services to the user’s habitual patterns.

We also have an interface role that acts as a gateway to the MAS. Thus, while the personal assistant accesses himself all information sources that he needs, the other systems like the backoffice send information to the MAS through the personal assistant. The backoffice and notification module are external actors to the MAS and are developed by our HERA partners.

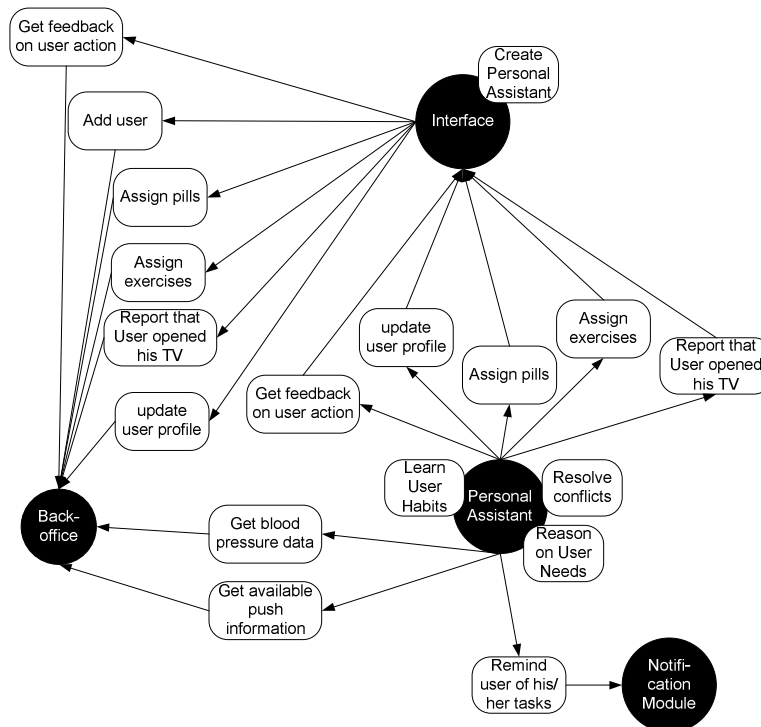


Fig. 1. The MAS System Actors Goals Model

An important analysis phase model that we define in ASEME is the System Role Model where the roles that are to be implemented as agents are refined and their specific capabilities and activities are defined. We use the Gaia operators as they were defined by Wooldridge et al. [16] for creating liveness formulas that define the process of the role. Briefly, A:B means that activity B is executed after activity A, A^ω means that activity A is executed forever (it restarts as soon as it finishes), A|B means that either activity A or activity B is executed, [A] means that activity A is optional and A||B means that activities A and B are executed in parallel. The left hand side term of the liveness formula is a capability, while the terms participating in the right hand side expression are activities or capabilities (if they appear on the left hand side of another liveness formula). The first formula has the role name on its left hand side.

PersonalAssistant	= initializeUserProfileStructure. initializeUserScheduleStructure. (newExercisePrescription ^ω newPillPrescription ^ω userOpenedTVSet ^ω serviceUser ^ω updateUserProfile ^ω)
userOpenedTVSet	= receiveUserOpenedTVInform. [invokeAvailableVideosInformService. [updateUserSchedule]]
updateUserSchedule	= resolveConflicts. updateUserScheduleStructure
resolveConflicts	= readUserScheduleStructure. (checkIfConflictsExist. reasonOnItemsPriorities. sortItems)+
updateUserProfile	= receiveUserProfileUpdateRequest. updateUserProfileStructure
serviceUser	= [checkUserExerciseResults]. [updateUserSchedule]. waitForUserScheduleNextItem. [reasonOnPillsQuantity]. [remindUserOfTasks]
checkUserExerciseResults	= invokeExerciseResultsService. checkIfUserDidHisExercises. [updateUserSchedule]
remindUserOfTasks	= invokeNotificationModule. [learnUserHabits]. getFeedbackOnUserAction
getFeedbackOnUserAction	= receiveUserActionInform. [updateUserSchedule]
learnUserHabits	= reasonOnUserAction. [updateUserSchedule]
newPillPrescription	= receiveNewPillPrescriptionRequest. updateUserSchedule
newExercisePrescription	= receiveNewExercisePrescriptionRequest. updateUserSchedule

Fig. 2. The PA role liveness formulas.

Having defined the SRM for the PA (see Figure 2) the next activity in the ASEME methodology is to define the functionality of each activity. This is done using the Functionality Table, which is presented in Figure 3 for the PA role. In the functionality table we have on the left hand side the agent capabilities, in the middle the activities that are used by each capability and on the right hand side the functionality that is used by each activity.

Thus, for developing the personal assistant (see Figure 3) we need to implement:

- two knowledge bases (one for assigning priority to conflicting user tasks and one for reasoning on the pills quantities in specific contexts)
- five ACL message receivers
- three algorithms (one for sorting items in the user's calendar, one for checking whether new calendar items conflict with existing items, and one for determining if the user has finished his exercises based on information retrieved from the HERA database)
- two global variables (user profile and user schedule) at the agent level
- three different web service clients, and,
- a statistical learning algorithm

ASEME caters for the transformation of the liveness formulas to a statechart that allows the designer to add conditions to the statechart transitions for controlling the execution of the activities. In the rest of this paper we will focus in the knowledge bases development.

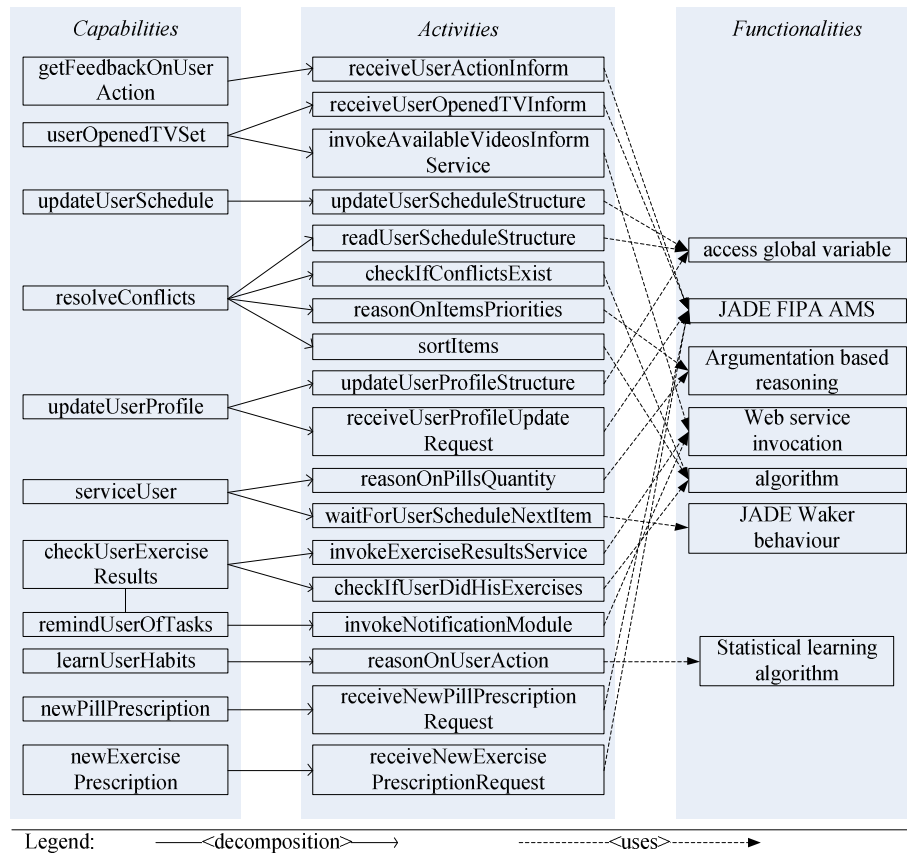


Fig. 3. The Personal Assistant role Functionality Table

4 Using Argumentation in HERA

We focus in two issues that are highly related to the assistance of people with mild or moderate Alzheimer's disease. The first suffer from frequent recent memory loss, particularly of recent conversations and events, while the second suffer from pervasive and persistent memory loss, confusion about current events, time and place [11]. Thus, it is important for the system to be able to help them in situations that they need to remember a special case for taking their pills or to schedule their tasks.

Reasoning On The Pills Quantities In Specific Contexts

When the user's scheduled time for taking pills arrives, the PA reasons on the quantity to be taken. The doctors are able to assign specific conditions when assigning pills to a patient using the HERA backoffice. For example, if the blood pressure exceeds a limit then he has to take two pills (while normally he is assigned one pill).

Moreover, there can be specific contexts that define specific situations like e.g. when the temperature is cold the blood pressure fluctuates, thus the readings are not so reliable.

Pills quantities can be modified based on basic rules identified and provided by doctors. For example if we want to work on blood pressure, in order to be able to use rules we need some threshold values provided by doctors. Here we will represent them with those expressions “Lower_Threshold” and “Upper_Threshold”. These upper and lower thresholds are not universal, they are defined by the doctors for each patient.

Note that we have some past measurements of the user either in a row or in the last two days. If all are up or down from a given threshold we have some rules for updating the pill quantity as prescribed by the doctor. Thus, the fact of what the new pill dosage will be if the lower threshold will be surpassed and what the pill dosage will be if an upper threshold is surpassed must be added.

If we have a specific context and precise information provided by doctors, thanks to the argumentation framework it is possible for us to transform it into an argumentation based decision making problem and find some solutions automatically. For the reader’s convenience, we present a simplified extract that demonstrates the achieved functionality. We define some predicates that are instantiated by the personal assistant `defaultPillDosage/2`, with the pill name and the dosage as a real number, `upperThresholdBreachedDosage/2`, with the pill name and the dosage in the case when the upper BP threshold is breached, `systolicBP/1`, containing the reading of the user’s blood pressure, `upperSystolicThreshold/2`, containing the BP threshold which must be surpassed so that a special dosage is proposed to the user, the `temperature/1` characterizing the temperature of the environment, and the `newPillQuantity/2` which after the reasoning phase contains the suggested dosage.

```

1. rule(r1, newPillQuantity(Pill, Dosage1),
2. [systolicBP(SBP), upperSystolicThreshold(Pill, Limit),
3. upperThresholdBreachedDosage(Pill, Dosage1), SBP >
4. Limit]).
5. rule(r2, newPillQuantity(Pill, Dosage2),
6. [defaultPillDosage(Pill, Dosage2)]).
7.
8. conflict(r1,r2).
9.
10. rule(pr1, prefer(r1,r2), []).
11. rule(pr2, prefer(r2,r1), [temperature(cold)]).
12.
13. rule(c1, prefer(pr2,pr1), []).

```

Rule r2 states that the new pill dosage is equal to the default. Rule r1 says that if the upper systolic pressure threshold is breached a new pill quantity is assigned. Line 8 characterizes these rules as conflicting and line 10 assigns a default higher priority to the r1 rule. However, the rule in line 11 assigns higher priority to rule r2 in the specific context where the temperature is cold (when BP tends to fluctuate). Finally, the rule in line 13 assigns a higher priority to rule pr2 over pr1.

Assigning Priority To Conflicting User Tasks

When an item is inserted in the user's agenda or is rescheduled the agent reasons on the priority of possibly conflicting tasks. Specifically, when the user has been assigned more than one tasks for the same time (e.g. by different caregivers) or that he has specific preferences (e.g. to watch a TV series at a particular time of day) the following priorities will hold (see the HERA system requirements [15]):

- Priority no1: take the assigned pills
- Priority no2: watch his favourite TV series
- Priority no3: engage with the cognitive reinforcement exercises
- Priority no4: engage with the physical reinforcement exercises

This knowledge base is the same for all users as the requirements that we got from the caretakers were such and is simpler than the previous one.

5 Conclusion

In this paper we have presented a multi-agent system part of the HERA system that is proposed for assuming Ambient Assisted Living functionalities providing at home services for people suffering from cognitive problems and more particularly from Alzheimer disease. We have presented several engineering aspects on how such a system can be designed by using the ASEME methodology along with elements concerning the reasoning mechanism used by some of the agents based on argumentation. We have thus shown that the latter allows agents to make decisions in situations where conflicting points of view, corresponding to doctors' opinions of different specialties and concerning patients having different health problems, must be taken into account. Our work has been functionally validated and is currently under user acceptance tests in the HERA trials.

The HERA system deployment will take place in two phases. In the first one it will be deployed in the medical center's premises for evaluation by the medical personnel and for controlled interaction with the patients and in the second phase it will be deployed inside the users' homes for final evaluation. The process for evaluating our system has been described in more detail in [11].

References

1. Becker M., Software Architecture Trends and Promising Technology for Ambient Assisted Living Systems, Dagstuhl Seminar Proceedings 07462, Assisted Living Systems-Models, Architectures and Engineering Approaches, 2008
2. Bravo, J., de Ipiña, D.L., Fuentes, C., Hervás, R., Peña, R., Vergara, M., Casero, G.: Enabling NFC Technology for Supporting Chronic Diseases: A Proposal for Alzheimer Caregivers. In: Ambient Intelligence (AmI-2008), LNCS, vol.5355, pp. 109--125. Springer Berlin / Heidelberg (2008)

3. Foster, D., McGregor, C., El-Masri, S. A Survey of Agent-Based Intelligent Decision Support Systems to Support Clinical Management and Research, in, *Proceedings MAS*BIOMED'05*, Utrecht, Netherlands, 2005
4. García, O., Tapia, D.I., Saavedra, A., Alonso R.S., García, I: ALZ-MAS 2.0; A Distributed Approach for Alzheimer Health Care. In: *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pp. 76--85. Springer (2008)
5. Kakas, A., Moraitis, P.: Argumentation Based Decision Making for Autonomous Agents. In: *Proc. 2nd Int'l Joint Conf. Autonomous Agents and Multi-Agent Systems (AAMAS 03)*, pp. 883--890. ACM (2003)
6. Kleinberger, T., Becker, M., Ras, E., Holzinger, A. & Müller, P.: Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces. In: *Universal Access to Ambient Interaction, LNCS*, vol. 4555, pp. 103--112. Springer Berlin / Heidelberg (2007)
7. Mahoney D.F., et al., In-home monitoring old persons with dementia: Ethical guidelines for technology research and development, *Alzheimer's & Dementia* 3 (2007) 217-226
8. Moraitis P, Spanoudakis N (2007) Argumentation-based Agent Interaction in an Ambient Intelligence Context. *IEEE Intell Syst* 22(6):84-93
9. Nehmer, J., Karshmer, A., Becker, M., Lamm, R.: Living Assistance Systems – An Ambient Intelligence Approach. In: *Proc. of the Int. Conf. on Software Engineering (ICSE)* (2006)
10. Prakken, H. and Vreeswijk, G.: Logics for Defeasible Argumentation, in: *Handbook of Philosophical Logic*, volume 4, D. Gabbay and F. Guenther, eds., Kluwer Academic Publishers, 2002, pp. 218-319
11. Spanoudakis N., Grabner B., Lymperopoulou O., Moser-Siegmeth V., Pantelopoulos S., Sakka P. and Moraitis P.: A Novel Architecture and Process for Ambient Assisted Living – The HERA approach. In: *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB 2010)*, Corfu, Greece, November 3-5, 2010
12. Spanoudakis N., Moraitis P.: Using ASEME Methodology for Model-Driven Agent Systems Development. In: *Agent-Oriented Software Engineering XI, LNCS*, vol. 6788, pp. 106--127. Springer Berlin / Heidelberg (2010)
13. Spanoudakis, N., Moraitis, P., Dimopoulos, Y.: Engineering an Agent-based Approach to Ambient Assisted Living. In: *AmI'09 Workshop on Interactions Techniques and Metaphors in Assistive Smart Environments (IntTech'09)*, Salzburg, Austria, November 18-21, (2009)
14. Spanoudakis N., Pendaraki K. and Beligiannis G.. Portfolio Construction Using Argumentation and Hybrid Evolutionary Forecasting Algorithms. *International Journal of Hybrid Intelligent Systems (IJHIS)*, Vol. 6, No. 4, 2009, pp. 231-243
15. The HERA Consortium: State-of-the-art and Requirements Analysis, Deliverable D2.1, <http://aal-hera.eu>
16. Wooldridge M, Jennings NR, Kinny D (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. *J Auton Agents and Multi-Agent Syst* 3(3):285-312