Using Agent Technology for Ambient Assisted Living

Nikolaos I. Spanoudakis[1] and Pavlos Moraitis[2]

[1]Applied Mathematics and Computers Laboratory – Technical University of Crete, Greece
nispanoudakis@isc.tuc.gr
[2]Laboratory of Informatics Paris Descartes (LIPADE) – Paris Descartes University, France
pavlos@mi.parisdescartes.fr

Abstract. The goal of this paper is to provide some arguments in favor of the use of agent technology in real word applications. It does so through the, as much as possible, detailed description of a multi-agent system in the context of a real world application in the domain of ambient assisted living. It discusses development issues involving the use of the platform JADE and integrating computational argumentation and web services. It also gives some feedback concerning the experience of engineering such systems, especially when integrated in a more complex system with other components, but also concerning the perception of real users when using the system in an everyday life context.

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

1 Introduction

This paper aims to present an agent-based architecture for addressing the non-trivial task [12] of engineering an Ambient Assisted Living (AAL) information system. AAL is about creating products and services that provide unobtrusive support for daily life based on context and the situation of the assisted person. It is currently one of the most important and well-funded research and development areas. The reason is the increasing average age of the total population, especially in the developed countries, that augments costs for the public, but also for individuals, for the care for the elderly people at home. As a consequence, the quality of life reduces for those people as their social skills diminish [8, 12].

The HERA project [24], co-funded by the AAL Joint Programme [17], built an AAL system to provide cost-effective specialized assisted living services for the elderly people suffering from Mild Cognitive Impairment (MCI), mild/moderate Alzheimer's Disease (AD), or other diseases (diabetes, cardiovascular) with identified risk factors, aiming to significantly improve the quality of their home life and extend its duration. HERA provided to its end users the following main categories of services:

- *Cognitive Exercises*, the end users play cognitive reinforcement games

adfa, p. 1, 2011.

- *Passive communication*, the end user can select (or receive a pro-active suggestion) to watch informative videos and news items related to his/her disease
- *Pill and Exercise reminders*
- *Reality orientation*, e.g. date and time are visible on screen
- *Blood pressure and weight monitoring*

Previous works in this area [3, 5] introduced agent based systems for AmI for assisting in taking care of people with Alzheimer but on one hand they required the aid of a nurse or caregiver and on the other hand they used RFID tags [5] or NFC technology [3]. Both technologies are used to identify the location of the user using radio frequency identification technologies that require the use of sensors in different places at home and the use of another sensor on the user's body for identifying his location. Another application for medicine usage management using agent technology which might concern different health problems is the one proposed by Hoogendoorn et al. [6]. This application explores and analyses possibilities to use automated devices such as an automated medicine box, servers and cell phones as non-human agents, in addition to human agents such as the patient and a supervisor doctor. What is missing from such systems and which we had to address in HERA is to increase the autonomy of the user, to automate the ambient assistance (without requiring the use of managers, caregivers or nurses), to act unobtrusively and to minimize the use of hardware, aiming to a commercial solution to home care for people suffering from dementia and having cognitive problems.

The HERA overall system service oriented architecture and evaluation process is presented in [13]. Briefly, to address the above challenges we used a combined agent- and service-oriented approach. For the agents decision making we chose argumentation [1], as it allows for decision making using conflicting knowledge, thus different experts can express their opinion that can be conflicting. Argumentation has been used successfully in the last years in similar situations, e.g. for deliberating over the needs of a user with a combination of impairments in an AmI application [11], or for group decision making in a more general setting in ambient intelligence environments [10]. The latter proposes a multi-agent (simulator) argumentation based system whose aim is to simulate group decision making processes. The use of the TV set and remote control for Human-Machine Interaction allowed for a quick learning curve for our users. We used a service oriented architecture based on web services that allowed the different sub-systems to be connected in a plug and play standardized way.

In this paper we focus in presenting the intelligent part of the HERA system, which is a multi-agent system. We aim to show that agent technology is useful for modeling real world AAL systems through presenting and discussing the architecture and use of technology for HERA.

2 System Architecture

The MAS is a module in the overall HERA system architecture [13] that interacts with the Human-Machine Interface (HMI) and the back-office. It participates in a service oriented architecture where every module can offer services over the web. The

MAS module is responsible for learning the user's habits, personalizing services such as the pill reminder and the passive communication, and for reminding him to do his daily tasks. These needs were identified during the requirements analysis phase [16].

The MAS module was developed using the Agent Systems Engineering Methodology (ASEME [14]), an Agent Oriented Software Engineering (AOSE) methodology. ASEME supports a modular agent design approach and introduces the concepts of intra- and inter-agent control. The first defines the agent's behavior by coordinating the different modules that implement his capabilities, while the latter defines the protocols that govern the coordination of the society of the agents. ASEME uses the Agent Modeling Language (AMOLA), which provides the syntax and semantics for creating models of multi-agent systems covering the analysis and design phases of a software development process. ASEME, uniquely among other AOSE methodologies, a) caters for the integration of the inter-agent control model to the intra-agent control model, and b) provides a set of graphical and transformation tools that not only allow for getting from requirements down to implementation but also initialize each new phase model based on the existing information in previous phases models (for a detailed discussion the reader can consult [14]). The analysis and development process for HERA using ASEME is presented in [9].

During the analysis phase, we defined an architecture for the MAS module using Interface and Personal Assistant agent types [15]. The *Interface* (INT) agent is used for connecting with other modules. Thus, the interface agent is the only one exposing the MAS services to the other modules. This agent gets all requests for service and delegates them to the interested personal assistant agent.

The *Personal Assistant* (PA) agent is the most complex role and it serves a registered user, stores and manages his/her profile and personal data and uses the requests' history in order to adapt the services to his/her habitual patterns. A different PA agent is active for each user. The PA is persistent and retains a schedule for the user's tasks. These tasks are assigned by the doctors that monitor the specific user, using the services' back-office system. The following tasks are achieved by the PA:

- 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. For example, if the blood pressure exceeds a limit then he has to take two pills (while normally he is assigned one pill)
- The agent learns the user's preferred time for the same type of activity (e.g. cognitive training). The algorithm searches for a pattern in recent acts, thus:
  - If the user did the same activity at the same time (differing by a few minutes) for this day of the week on the last two weeks then the agent proposes this time.
  - Else if the user did the same activity at the same hour on the last two days, then the agent proposes this time
- When an item is inserted 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 he has specific preferences (e.g. to watch a TV series at a particular time of day) the following priorities will hold:

- Priority no1: take the assigned pills
  - Priority no2: watch his favorite TV series
  - Priority no3: engage with the cognitive reinforcement exercises
  - Priority no4: engage with the physical reinforcement exercises
- Once a week the agent selects a piece of information (video) to propose to the user.

For developing the MAS module, we chose the Java Agent Development Framework (JADE), as it is the most popular agent platform [2] that complies with the FIPA [18] standards, and as the adoption of ASEME allows for semi-automatic code generation.

For the agents decision making we chose argumentation as it is very well suited for implementing decision making mechanisms dealing with the dynamic nature of possibly conflicting actions due to different situations or contexts. The framework we used [7] is based on object level arguments representing the decision policies and then it uses 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. This framework has been applied in a successful way in different applications (see e.g. [11]) involving similar scenarios of decision making and it is supported by the Prolog-based Gorgias open source software [19]. The different components participating in the HERA MAS server are identified in the architecture diagram presented in Figure 1. All the agents communicate using FIPA Agent Communication Language (ACL)-based communication protocols and messages.

For exposing interfaces of services offered by the MAS, we used the Web Service Integration Gateway (WSIG) JADE add-on, which provides support for invocation of JADE agent services from Web service clients. The WSIG servlet exposes the web services and the WSIG agent transforms them to ACL messages using the MAS ontology and sends them to the interface agent. The latter decides if he needs to create a new personal assistant or to forward the request to one of the existing personal assistants.

For defining the interfaces of the MAS module with the other modules and also the inter-agent messages we defined an ontology (i.e. a list of concepts). The ontology was developed using the bean generator add-on of the Protégé open source ontology editor [22]. The process involves the generation of a suitable WSDL (Web Services Definition Language) file for each service-description registered with the Directory Facilitator (DF) agent.

Regarding security, the MAS module uses only the user's username in all transactions. Thus, the MAS module is never aware of the user identification data other than those of the username. Security in communication is achieved through the use of secure layer protocols (e.g. HTTPS) and through allowing only the HERA back-office to access the web services through the firewall.

The JADE Persistence add-on allows for saving the state of agents (recover user profiles and learned data in the case that the system crashes), using the HSQL relational database. Hibernate [20] was used for automating the transformation of the agents' data structures to relational database tables.

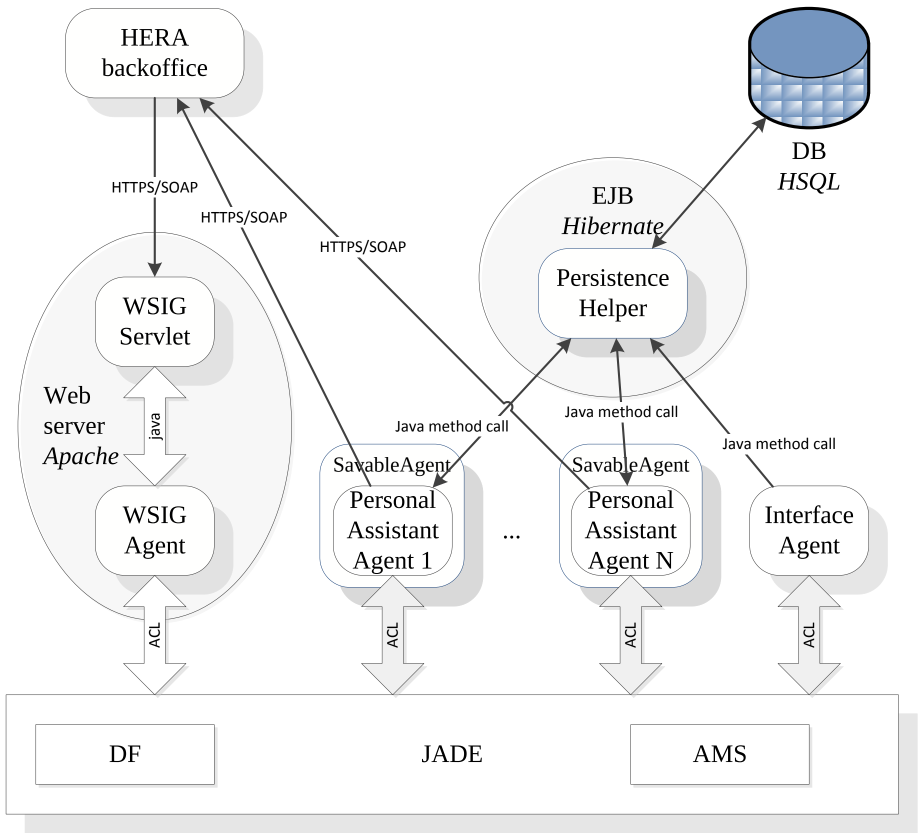

Fig. 1. HERA MAS Architecture

3 Validation and Evaluation

For validating the MAS server functionality, we defined specific scenarios of tests in four levels. Firstly, we validated the learning algorithms and the Prolog rule base against test data. Then, in a second level, we tested the individual agent types for their conformance in the defined protocols and tested their algorithms again using specific ACL messages. In the third level we tested the functionality of the MAS using a web services client to send specific service requests. Finally, after integrating all the HERA components in the application platform, we tested complete predefined usage scenarios using the HMI. Then the system was evaluated by the end users with the objective to assess (with metrics) the added value of HERA. The Process of Evaluation of Software Products [4] has been used for evaluating the HERA services.

The HERA services related to the MAS module were put under trial in Greece. The pilot operation took place at the Hygeia hospital [21]. We focused in two categories of users: the end-users (who use the HERA services), and the Medical Personnel (who configure the HERA services and assess the end users' progress). A total of 30 end-users (10 healthy elderly, 8 suffering from MCI, 8 from mild AD, and 4 from moderate AD) were selected to participate in the project trials phase, along with 10 medical experts. The equipment installed included a set-top-box (small PC), a TV-set and a remote control. Both end users and medical experts were trained at the hospital. Then they completed questionnaires. That was the end of the first phase.

For the second phase two helpdesks were setup. The main one was the 24/7 helpdesk at Hygeia hospital, which was equipped with trouble shooting steps to be followed at each kind of problem. If the problem was not resolved, then it was forwarded to the Singular Logic [23] helpdesk either through email or voice mail. Then, the equipment was installed to the end-users' homes. They were also equipped with a 3G internet providing stick if the end-users did not have an ADSL connection. The HERA equipment stayed at the user's home for a minimum period of 15 days. Finally, the MAS module was connected to the HERA system for adapting the service to each user's habits and needs. An Hygeia representative from the memory clinic visited the user to get him/her to fill the questionnaire. As soon as a doctor's users all finished their trial usage, the doctor also filled in the questionnaire. The evaluation criteria reflected the way of achieving the objectives. They were the following:

- Performance (C1), measures the capability of the system to produce valid and accurate results.
- Usability (C2), measures the satisfaction of the user with regard to his experience in using the system and the ease of achieving his tasks.
- Flexibility (C3), refers to the ease of troubleshooting and of moving from one service to another.
- Security and Trust (C4), refers to the user perception of whether sensitive data are securely handled and remain confidential, also to whether the user trusts the system when it proposes a course of action.

Figure 2 compares the system's performance during the second trial phase to its performance during the first phase for the end-users group. In the first phase, the end users saw a substantial performance gap in the Performance criterion. This is reasonable considering that, during the trials at the hospital the bandwidth was very low resulting in problems with viewing videos. Usability and Security and Trust criteria performance is considered satisfactory (both over 80%). However, the users showed skepticism on whether they are able to use this system at home (see the performance of the C3-Flexibility criterion). This was mainly due to the fact that the level of cognitive impairment directly affects the end users' views on their ability to use HERA at home: the wider the extent of cognitive impairment, the lower the perceived ability to use the system without assistance. Additionally, the vast majority of the end users (as most Greek elderly) were technologically illiterate and exhibited fear, doubt and uncertainty when asked to use technology-based services, even if, in the case of HERA, they were offered through a regular TV set, which they were familiar with.

Looking at the phase two bars, the reader can notice the substantial improvement in the Performance criterion. One of the reasons was the software agents that assisted the users as in the first phase there was no learning or personalization. The usage of the 3G USB sticks and the ADSL broadband connections helped in remedying the low bandwidth situation at the hospital. Furthermore, the original skepticism that the users showed on whether they will be able to use this system at home (see the performance of the Flexibility criterion) was finally dispersed. We believe that the use of the familiar remote control along with the setup of the help center in the Hygeia hospital helped substantially, since the end users at home did not need to talk to technicians if they had a problem, but a person from the hospital.

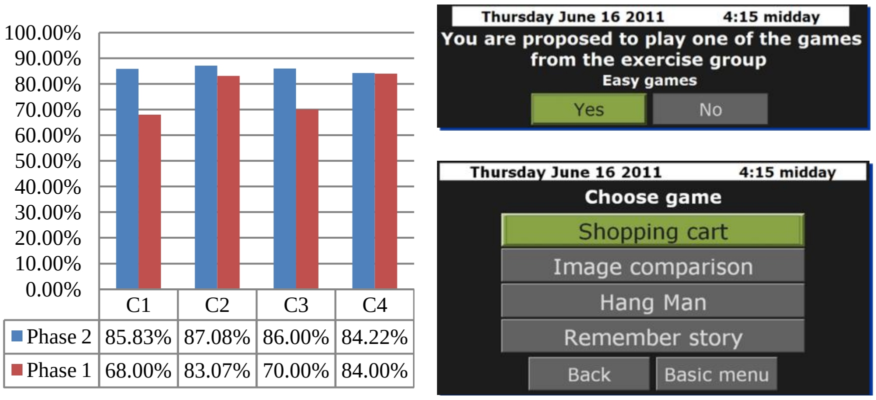

Fig. 2. The 1st and 2nd trial phases system performance (end users) on the left side, and two screenshots of the system in use on the right side: a) the user is reminded to play cognitive games (top), and, b) the user selects one game to play (bottom).

4 Discussion and Conclusion

Our goal in this paper was to provide the reader with information on how to engineer a real-world agent-based system and integrate it to a more general Service-oriented Architecture (in this case using the web services paradigm). We showed how to combine the WSIG and Persistence add-ons of JADE in an architecture with the interface and personal assistant agent types.

We also provided some state of the art information on agent-based systems for the ambient assisted living (AAL) domain. Related to that, in HERA we managed to increase the autonomy of the user, to automate the ambient assistance (without requiring the use of managers, caregivers or nurses), to act unobtrusively and to minimize the use of hardware, achieving a commercial solution for home care for people suffering from dementia and having cognitive problems.

We believe that the presented information can be exploited in a wider range of application domains as the agent types that we used are widely used and have been proposed as generic agent types by the agent technology community.

5 References

1. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence. 171, 10-15, 619–641 (2007).

2. Bordini, R.H. et al.: A Survey of Programming Languages and Platforms for Multi-Agent Systems. Informatica. 30, 33–44 (2006).
3. Bravo, J. et al.: Enabling NFC Technology for Supporting Chronic Diseases: A Proposal for Alzheimer Caregivers. In: Aarts, E. et al. (eds.) Ambient Intelligence. pp. 109–125 Springer Berlin Heidelberg, Berlin, Heidelberg (2008).
4. Colombo, R., Guerra, A.: The Evaluation Method for Software Product. Proceedings of the 15th International Conference on Software & Systems Engineering & Applications. , Paris (2002).
5. Corchado, J. et al.: Intelligent environment for monitoring Alzheimer patients, agent technology for health care. Decision Support Systems. 44, 2, 382–396 (2008).
6. Hoogendoorn, M. et al.: Formal Design and Simulation of an Ambient Multi-agent System Model for Medicine Usage Management. In: Mühlhäuser, M. et al. (eds.) Constructing Ambient Intelligence. pp. 207–217 Springer, Berlin, Heidelberg (2008).
7. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. Proceedings of the second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'03). pp. 883–890 ACM Press, New York, USA (2003).
8. Kleinberger, T. et al.: Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces. In: Stephanidis, C. (ed.) Universal Access in Human-Computer Interaction. Ambient Interaction. pp. 103–112 Springer Berlin Heidelberg, Berlin, Heidelberg (2007).
9. Marcais, J. et al.: Using Argumentation for Ambient Assisted Living. In: Iliadis, L. et al. (eds.) Artificial Intelligence Applications and Innovations. Springer Berlin Heidelberg, Berlin, Heidelberg (2011).
10. Marreiros, G. et al.: Argumentation-Based Decision Making in Ambient Intelligence Environments. In: Neves, J. et al. (eds.) Progress in Artificial Intelligence. pp. 309–322 Springer, Berlin, Heidelberg (2007).
11. Moraitis, P., Spanoudakis, N.: Argumentation-Based Agent Interaction in an Ambient-Intelligence Context. IEEE Intelligent Systems. 22, 6, 84–93 (2007).
12. Nehmer, J. et al.: Living assistance systems: an ambient intelligence approach. Proceedings of the 28th international conference on Software engineering. pp. 43–50 , Shanghai, China (2006).
13. Spanoudakis, N. et al.: A novel architecture and process for Ambient Assisted Living - the HERA approach. Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine. IEEE (2010).
14. Spanoudakis, N., Moraitis, P.: Using ASEME Methodology for Model-Driven Agent Systems Development. In: Weyns, D. and Gleizes, M.-P. (eds.) Agent-Oriented Software Engineering XI. pp. 106–127 Springer-Verlag, Berlin, Heidelberg (2011).
15. Sycara, K., Zeng, D.: Coordination of Multiple Intelligent Software Agents. International Journal of Cooperative Information Systems. 05, 2-3, 181–211 (1996).
16. The HERA Consortium: Project Deliverable D2.1: State-of-the-art and Requirements Analysis. http://w3.mi.parisdescartes.fr/hera (2010).
17. Ambient Assisted Living Joint Programme, http://www.aal-europe.eu.
18. Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org.
19. Gorgias general argumentation framework, http://www.cs.ucy.ac.cy/~nkd/gorgias.
20. Hibernate Object/Relational persistence and query service, http://www.hibernate.org.
21. Hygeia hospital, http://www.hygeia.gr.
22. OntologyBeanGenerator, http://protegewiki.stanford.edu/wiki/OntologyBeanGenerator.
23. SingularLogic Software and Integrated IT Solutions Group, http://www.singularlogic.eu.
24. The HERA project (AAL-2008-1-079), http://w3.mi.parisdescartes.fr/hera.