A Hybrid Multi Agent System Architecture for Distributed Supervision of Chronic Patients in the eHealth Setting

Olivier A. Blanson Henkemans[1,2]
Stefano Bonacina[3]
Nicola Cappiello[3]
Charles A.P.G. van der Mast[1]
Mark. A. Neerincx[1,2]
Francesco Pinciroli[3]

[1]Delft University of Technology
Mekelweg 4
2628 CD Delft
E-mail:
{O.A.BlansonHenkemans|
C.A.P.G.vanderMast} @TUDelft.nl

[2]TNO
Kampweg 5
3769 GZ Soesterberg
E-mail:
Mark.Neerincx@TNO.nl

[3]Politecnico di Milano
Piazza Leonardo da Vinci, 32
20133 Milano
E-mail:
{Nicola.Cappielo|
Francesco.Pinciroli|
Stefano.Bonacina} @Polimi.it

ABSTRACT

eHealth, the use of Information and Communication Technology (ICT) in the health sector, enhances today's health care environment. In particular, the use of Multi Agent System (MAS) technology, an aspect of ICT, can further contribute to the improvement of health care. Exceptional integration of this technology requires a Cognitive Engineering (CE) approach. Implying that the design and implementation of the architecture is done incrementally and using multiple distributed agents, which are facilitated by easy data entry, management and verification by all involved actors. Consequently, we designed a hybrid agent architecture, consisting of multiple distributed agents and a Virtual Personal Assistant that supervises patients' self-care, with chronic illness, and tested it in a laboratory setting according to the Cognitive Engineering approach. Results showed that the architecture designed meets with our requirements and the functionalities are comprehensible by the involved users. We expect that the incremental character of the developed hybrid architecture enables further development and could be applicable for the supervision of a variety of chronic diseases in the eHealth setting.

KEYWORDS

eHealth, Multi Agents System, Virtual Personal Assistants, Hybrid Agent Architecture, Cognitive Engineering,

INTRODUCTION

eHealth, the application of Information and Communication Technologies (ICT) in the health sector, is radically changing today's health care (Curry et al., 2002). Due to the combination of decreasing health care costs, exponentially increasing network connection speeds, and the suitability of eHealth to support patients' decisions-making and "supervised autonomy", the application of eHealth has the capacity to considerably increase the availability of self-care options (Leventhal et al., 2004). Self-care is defined by Bhuyan as activities individuals, families, and communities undertake with the intention of enhancing health, preventing disease, limiting illness, and restoring health, which in turn can improve a patient's lifestyle, medical adherence, and future health outcome (Bhuyan, 2004). An example of eHealth is the use of ICT in diabetes care. Rule-based reasoning is combined with Multi Agent Systems (MAS) technology (Sycara, 2001; Wooldridge, 2002) using mathematical models of blood glucose regulation for the identification of problems and treatment generation. Subsequently, treatment can be prescribed, including insulin therapy, diet and physical exercise (Blanson Henkemans, 2006; Haan et al., 2005).

For people with a chronic disease, eHealth solutions using Multi Agent System (MAS) technology can provide the following functionalities, concerning the personalized assistance for multiple distributed actors, e.g., patients, physicians, and other medical specialists (Xiao et al., 2006; Tonino et al., 2002; Lindenberg et al. 2003; European Commission, 2003). These solutions include improving relationships between clients and caregivers, detecting adverse trends in health proactively, stimulating a patient's motivation and bring about behavioral change. The latter is required for effective self-care, patient's quality of life, information sharing with the involved medical specialists, and productive and low-cost self-care.

To provide the above mentioned functionalities, we designed an architecture, consisting of Multi Agent System (MAS) technology. The design took place in the framework of the SuperAssist project, a collaboration between Delft University of Technology, TNO and LUMC (Haan et al., 2005). To facilitate computer-supported task performance by increasing insight into the cognitive factors, such as reasoning, quantitative knowledge, and short-term memory (Carroll, 1993) of human-computer interaction, we applied a Cognitive Engineering (CE) approach (Neerincx & Lindenberg, 2007). First, we wanted an architecture that enables designing and implementing incrementally, implying that the designed MAS architecture should be suitable for further expansion, concerning distributed actors, both human and machine, tools, e.g., medical devices, and data sources, e.g., electronic patient records, and should be adaptable to

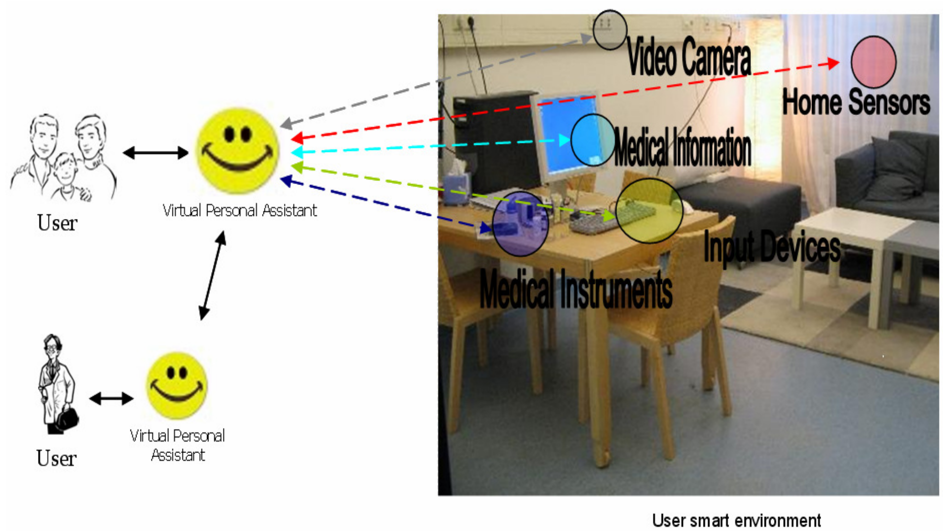

Figure 1: Multi-Agent System (MAS) working in a smart environment and interact with the users through their Virtual Personal Assistants (VPA)

diverse situations, which too are also dynamic with time. Consequently, the intelligence of the system should be added step by step. Second, we wanted multiple distributed agents that could act both as independent intelligent actors in the network managing the data and also as assistants for the involved human actors, e.g., patients, medical specialists, and technical specialists, by supporting them in their complex task environment. Third, we wanted a system that enables easy data manipulation, including data entry, management and review, because all actors involved should be able to perform each of these actions.

Several projects (e.g., Bellazzi et al. 2003; Camarina-Maots et al., 2003; DbMotion, 2005) offer well-founded agent architectures, but also have a number of shortcomings, in terms of the requirements mentioned above (Table 1). In short, the existing architectures apply a one-side centralized or decentralized approach. This implies that the data is either collected from the actors in a network, converted to a single format, and stored in one database that serves all its participants, or that the actors maintain ownership of their data, which then has to be retrieved by other actors.

To tackle these disadvantages, we designed a hybrid architecture, consisting of a decentralized system with a central node. This node can manage data that the actors decide to synchronize, e.g., medical test results and the medical specialists' calendar. The same node can contain a web server, a communication server and other central systems that need a centralized structure. The hybrid architecture enables the use of multiple distributed agents that both act as independent intelligent actors in the network that manage the data and as assistants to the users. Furthermore, this hybrid approach makes the architecture suitable for further expansion and adaptation.

In this paper, we will discuss the hybrid architecture designed to address the shortcomings mentioned. In addition, we will report the results of a qualitative experiment conducted to test the hybrid architecture's functional capacity and whether or not it fulfils the requirements of the Cognitive Engineering (CE) approach, concerning designing and implementing the architecture incrementally, using multiple distributed agents, and easy data entry, management and checking by all the involved actors. Finally, we will discuss the implications of the results.

Table 1: Disadvantages of centralized and decentralized agent architectures

Centralized architecture	**Decentralized architecture**
Constrained flexibility in distributed database management	Difficult to integrate and interoperate different platforms
High dependency on difficult accessible internal network	Distributed data sources and agents are required to be online for data synchronization
All the computation must take place centrally	Indistinctness of who is managing the data
	Prone to data redundancy

2. A HYBRID ARCHITECTURE IN AN FOR THE SUPERIVISON OF DIABETES PATIENTS

The main goal was to design a hybrid Multi Agent System architecture suitable for the supervision of diabetes patients in an eHealth setting according the Cognitive Engineering approach. In our architecture, agents are working in the background providing ambient intelligence to the users who reside in the smart environment. A Virtual Personal Assistant (VPA) acts as mediator between the users and agents active in the smart environment (Lindenberg et al. 2003; Maes, 1998; Grill et al., 2005). In the intelligent environment, the agents communicate with each other and receive information through sensors placed in the environment. The multiple agents acquire information through sensors, their behavior or their communication. The VPA shares data with the agents and interacts with the user. To test our architecture, we

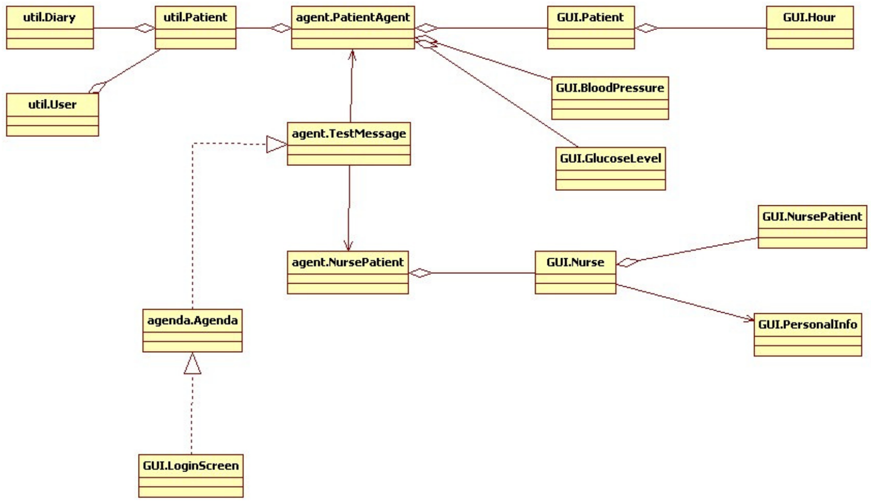

Figure 3: UML case diagram, diagram Virtual Personal Assistant (VPA) software

implemented a hybrid architecture, consisting of Multi-Agent System (MAS) technology and Virtual Personal Assistants (VPA) (Figure 1). Both the patient and the medical specialist have a VPA enabling them to interact with the Multi-Agent System. The Multi Agent System receives data derived from the patient's electronic patient record, electronic diabetes diary, and domestic medical instruments, e.g., glucometer. Based on this data, the patient's Virtual Personal Assistant interacts with the patient, e.g., about his or her health status and diseases. In addition, it updates the medical specialist's VPA, which in turn informs the medical specialist of the patient situation and supports managing basic medical data.

All the data that appears in the Diary is also stored in a database. The database is initialized with MySQL and managed by the java file DatabaseMediator.java of the library agendaDb.db (Figure 2). In the Unified Modeling Language (UML) case diagram (Figure 3), we represent the main tasks that every actor can perform with the system. There are four categories: objects needed for the actual communication with the database (agendaDb.db), objects needed to store data (agendaDb.util), objects needed for the graphical ser interface (agendaDb.GUI), and objects needed for the implementation of the agents (agendaDb.agent).

The software project is structured in five packages according to the general use of the classes contained. The agendaDb.agenda contains only one class system, which verifies if the user is in the database, and according with his group initiates a TestMessage object.

2.1 Agent Communication and Database

To enable optimal functionality, different agents run simultaneously on the patient and medical specialist's computers and communicate with each other. We have implemented two agents using the JADE platform, i.e., the *PatientAgent* on the patient's computer, and the *NurseAgent* on the medical specialist's computer. The agent communication takes place by sending messages from both sides. In order to recognize the different kind of messages, the behavior of the agents changes according to the prefix of the message. If the message starts with "start", the chat becomes active; if it starts with "chat", the chat window will show the message sent by the medical specialist. The third type of prefix is "response" followed by the name of the measurement, i.e. blood pressure or glucose level. Also, the *PatientAgent* checks how the new measurements influence the status of the patient.

2.2 Virtual Personal Assistant Activities

The Virtual Personal Assistants enable the involved human actors to interact with the different agents in the environment. This is realized through the performance of three main activities: data entry, formulate policies and make recommendations.

The medical specialist enters data regarding patient information, e.g., demographic data, medical history, and clinical diabetes information. The patient keeps track of self-care tasks in a personal electronic diabetes diary, including current mood, exercises performed, meals consumed, medication taken, and blood glucose measurement results. The data are entered by the patient and the medical specialist, through their Virtual Personal Assistants (VPAs) into a MySQL database.

The Virtual Personal Assistant gives short- and long-term suggestions, or so-called policies. This is based on the data in the electronic diabetes diary and electronic patient record. The VPA predominantly determines its prescribed policies after monitoring the results of the glucose test results, due to its importance in diabetes care.

The connection of the VPAs with the database is done with the Java Database Connectivity (JDBC) platform and the Application Program Interface (API), a programming interface allowing external access to the database manipulation and update commands. It allows the integration of SQL calls into a general programming environment by providing library routines, which interfaces with the database.

In theory, all actors have their database and the database is synchronized with a central node. In our architecture, the database, containing the electronic patient record and the electronic diabetes diary, is located on the patient's side. Thus when the patient is online, the database and current patient data is accessible by the medical specialist. When the patient is offline, the medical specialist has access to the patient's data retrieved the last time the patient was online and the databases were synchronized.

2.3 User Interface

Both the patient and medical specialist have an interface in which their Virtual Personal Assistants are integrated. In this section we will shortly discuss the interfaces. The patient's interface consists of six fields (Figure 4):

1. An electronic diabetes diary in which the patient logs the performed self-care tasks;
2. Access to the Electronic Patient Record (EPR);
3. Entering medical test results and viewing old medical results;
4. A "traffic light" indicating current health status based on diary and last measurement results (green: healthy, orange: be aware, red: alert!);
5. A chat service to communicate with the remote medical specialist, and;
6. A frame to communicate with the Virtual Personal Assistant.

The medical specialist also has an interface used for remote monitoring of the patients. This interface can be divided in three main parts (Figure 5):

1. List of the patients in the medical specialist's folder with the current health status represented by the accompanying traffic light.
2. Access to Patient Data Management, and;
3. A chat service to communicate with the patient.

- In the UML case diagram (Figure 3), the interface classes (GUI.x) are represented by:
- *GUI.Patient*: The main window of the interface from the patient side;
- *GUI.Nurse*: The main window of the interface from the medical specialist side;
- *GUI.BloodPressure* and *GUI.GlucoseLevel* are linked to *agent.PatientAgent* through the ActionEvent functions. When the user pushes the button '*Measurement done: update agenda*' the Patient Agent is announced to start the communication with the NurseAgent;
- *Util package*: Contains the info that is later stored in the database or retrieved from the database. Also in the *util.Diary* class, the thresholds for the measurements are verified and according to the new status the *GUI.Patient* shows the correct frames of the Virtual Personal Assistant.

3. METHOD: A QUALITATIVE EXPERIMENT

To test the developed hybrid architecture, consisting of Multi Agent System technology and the Virtual Personal

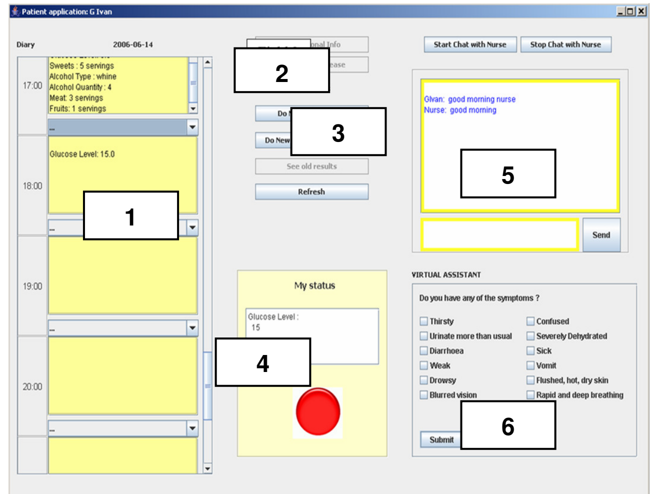

Figure 4: Patient Interface, containing 1. Electronic Diabetes Diary, 2. EPR access, 3. Retrieving tests results, 4. Traffic light, 5. Chat service, and 6. VPA communication frame.

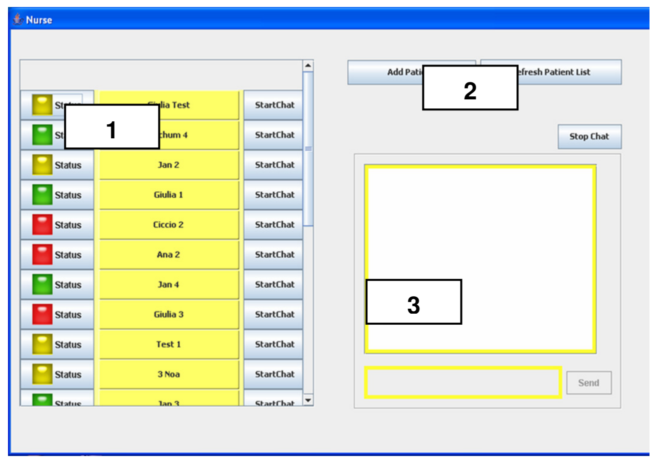

Figure 5: Medical Specialist Interface, containing 1. Patient List, 2. Access to Patient Data Management, and 3. Chat service.

Assistants, according the Cognitive Engineering requirements, we conducted a qualitative experiment. We invited eleven participants (nine students and two older adults) at Delft University of Technology (DUT) and TNO's Experience labs to evaluate the system. These labs enable testing prototypes of new technologies by offering a comfortable domestic atmosphere and encourage natural behavior in an experimental setting, while contextualizing the experience (Blanson Henkemans et al., 2007).

Because we wanted to evaluate the system before actually applying it in the field, we tested the system in the lab setting and also asked the participants to play the role of the patient according to predefined scenarios. A scenario is a description that contains actors, background information on the actors, their assumptions about their environment, actors' goals or objectives, and sequences of actions and events (Rosson & Carroll, 2001). Following the scenario-based design method, we used these scenarios as a general representation throughout the entire system lifecycle (analysis, design, prototyping and evaluation).

During the experiment, we observed the participant remotely with both an overview camera and a close-up camera. In addition, we had access to the medical specialist interface and a replica of the patient's interface. Finally, we logged the communication between the two agents. Consequently, we received a good overview of the tasks performed by the participant, the communication between the agents, and the functionality of the two Virtual Personal Assistants. Finally, we surveyed the participants' opinion on the VPA's usability.

4. RESULTS

When both Virtual Personal Assistants were online, they could correctly add and retrieve data to and from the database, containing the electronic patient record (EPR) and the electronic diabetes dairy. In the case of a health critical situation or when the patient sent a message to the medical specialist, the two VPAs could communicate with each other directly. The interaction and communication processes are illustrated in Figure 5. In practice, the patient logged self-care tasks in the electronic diary and based on the data, the patient's VPA made inferences about the patient's health status and gave feedback. The new data, based on the interaction between the patient and its VPA, was then sent to the database and retrievable by the medical specialist's VPA.
During the experiment, we observed that the data were accessible and manageable by the authorized actors. The patient and its Virtual Personal Assistant could edit the electronic diabetes dairy and view the electronic patient record. The medical specialist could add new patients to the database, edit the electronic patient record, and view the patient's diabetes diary and current health status. The medical specialist's VPA could view the patient electronic diabetes diary, electronic patient record and current health status.
When the patient's VPA was offline, it could still give comment on the diary entry and subsequently add the new data to the database. When the patient's VPA was online again, the medical specialist's VPA could synchronize the new data.
The patient's VPA reacted accurately and quickly to the newly added data of the patient electronic diary independent of the type of task or the health situation of the patient. The data was correctly added to the database and retrieved by the medical specialist's VPA. Also, the communication between the two VPAs was instantaneous. According to the usability survey, on a scale of 1 through 7 (7 being very usable) participants rated the usability of the assistant a 6.5.

5. DISCUSSION

The results depict that the designed hybrid architecture meets with our SuperAssist project requirements, concerning designing and implementing the architecture incrementally, using multiple distributed agents, and easy data entry, management and checking by all the involved actors. These functionalities work adequately and are comprehensible by the involved users through a usable Virtual Personal Assistant.
However, the system needs improvement on several aspects. One aspect we need to improve is the agents' intelligence

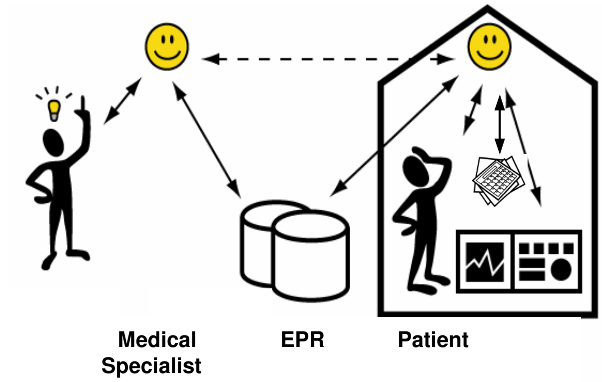

Figure 5: Virtual Personal Assistants for the supervision of patient's self-care

and awareness of processes. Currently, the Virtual Personal Assistants only monitor newly added data and do not take into consideration existing data. For example, with diabetes patients, but also with other chronic patients, e.g., people with asthma, the course of their health is as important as their current health state. Another aspect is the necessity to elaborate on privacy and security issues, such as risks of corporate espionage, consumer/personal privacy validation, and location privacy [Xiao et al., 2006]. If we want to deploy the hybrid architecture in real practice, we will have to further study the problems and restrictions these issues pose. Finally, we suggest adding a technical specialist to the architecture. The technical specialist, also equipped with a Virtual Personal Assistant, could help maintain and troubleshoot technical problems with both the system itself and the domestic medical instruments used by the patient.
In conclusion, we expect that the expansive character of the developed hybrid architecture facilitates implementing these improvements. In addition, it could be applicable for the supervision of a variety of chronic diseases in the eHealth setting.

ACKNOWLEDGMENTS

We would like to thank Oana Dragomyr for her essential work on the system. The SuperAssist project is partially funded by SenterNovem IOP-MMI, a program of the Dutch Ministry of Economic Affairs.

REFERENCES

Bellazzi R., Carson E. R., Cobelli C., Hernando E., Gomez E. K., & Nabih-Kamel-Boulos M. (2001). Merging telemedicine with knowledge management: the M2DM project. In: IEEE editor. *Proceedings of 23rd Annual EMBS International Conference of the IEEE*, Istanbul, Turkey, 4, pp. 4117-4120.

Bhuyan, K.K. (2004). Health promotion through self-care and community participation: Elements of a proposed programme in the developing countries. *BMC Public Health*, 4(11).

Blanson Henkemans, O.A., Neerincx, M.A., Lindenberg, J., & Mast, C.A.P.G van der (2006). SuperAssist: Supervision of Patient Self-Care and Medical Adherence. *IEA 2006 Conference,* July 10 - 14 , 2006, Maastricht, The Netherlands.

Blanson Henkemans, O.A. Caine, K.E., Rogers, W.A., Fisk, A.D. Neerincx, M.A., & Ruyter, B. de. (2007). Medical Monitoring for Independent Living: User-Centered Smart Home

Technologies for Older Adults. Accepted for the *Med-e-Tel 2007*, April 18 - 20, Luxemburg, Luxemburg.

Camarinha-Matos L.M., Octavio Castolo L., & Rosas J. (2003) A multi-agent based platform for virtual communities in elderly care. Emerging Technologies and Factory Automation; ETFA' 2003. *Proceedings of 9th IEEE International Conference on Emerging Technologies and Factory Automation*; Lisbon (P), September 16-19; 3, pp. 421-428.

Carroll, J.B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*, New York: Cambridge University Press, 1993.

Curry, R. G., Tinoco, M. T., & Wardle (2002). *The Use of Information and Communication (ICT) to Support Independent Living for Older and Disabled People*. Retrieved January, 2007 from http://www.rehabtool.com/forum/discussions/ictuk02.pdf.

DbMotion (2005). *The dbMotion Solution. A White Paper to access medical information*. Retrieved February, 2007 from http://www.ness.com/NR/rdonlyres/BCC63AB7-8C6D-402E-A923-7ED8497937FE/4757/WhitePaperFinalDec05.pdf

European Commission (2003). *The Health Status of the European Union: Narrowing the Health Gap. Office for Official Publications of the European Communities*. Luxembourg. Retrieved February, 2007 from http://ec.europa.eu/health/ph_information/documents/health_status_en.pdf

Grill, T. Khalil Ibrahim, I, & Kotsis, G. (2005). Agent interaction in ambient intelligent environments, *Proceedings of 13th DPS workshop*, November 29.-December 2, 2005, Okinawa - Japan.

De Haan G. de, Blanson Henkemans O.A., Ahluwalia A. (2005). Personal assistants for healthcare treatment at home. In: Marmaris N, Kontogiannis T, Nathanael D, editors. *Proceedings of European Association of Cognitive Ergonomics (EACE) 2005,* September 29-October 1, 2005, Crete (GR). pp. 217-223.

Leventhal H., Halm E., Horowitz C., Leventhal E. & Ozakinci G. (2004). Living with chronic illness: A contextualized, self-regulation approach. In: Sutton S, Baum A, Johnston M, editors. *The SAGE Handbook of Health Psychology*. London (UK): Sage Publications, 2004. p. 159-194.

Lindenberg J., Nagata S., & Neerincx M.A.. (2003). Personal assistant for on Line services: addressing human factors, human centered computing: cognitive, social and ergonomic aspects. *Proceedings of Human Computer Interaction International Conference*, 2003, Mahwah, NJ: Lawrence Erlbaum.

Maes P. (1998). Agents that Reduce Work and Information Overload. In: Maybury MT, Wahlster W: *Readings in Intelligent User Interfaces*, Morgan Kaufmann, 1998, pp.525-536.

Neerincx, M.A. & Lindenberg, J. (in press). Situated cognitive engineering for complex task environments. In: Schraagen, J.M. (Ed.), *Natural Decision Making & Macrocognition*. (will appear in 2007). Ashley.

Rosson M.B. & Carroll J.M. (2001) *Usability Engineering: Scenario-Based Development of Human Computer Interaction*. San Francisco, CA: Morgan Kaufmann.

Sycara K.P. (2001). Multiagent system: a modern approach to distributed artificial intelligence. *AI Magazine*, 22(2), pp. 105-108.

Tonino J.F.M., Bos A., Weerdt M.M., & Witteveen C. (2002). Replanning by Revision in Collective Agent-Based Systems. *Artificial Intelligence Journal,* 2002, 142(2), pp. 121-145.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*, John Wiley & Sons Ltd, 2002

Xiao, Y., Shen, X., Sun, B., & Lin, C. (2006). Security and privacy in RFID and applications in telemedicine. *IEEE Communications Magazine*. 2006, 44(4), pp. 64-72.

BIOGRAPHY

OLIVIER BLANSON HENKEMANS is PhD student in Human-Machine Interaction at Delft University of Technology and is also affiliated to TNO Human Factors. The focus of his research is on personal computer assistant supervision of older diabetes patients' self-care and context aware and adaptive user-assistant interaction.

MARK NEERINCX is head of the Intelligent Interface group at TNO Human Factors, and professor in Man-Machine Interaction at the Delft University of Technology. He has extensive experience in applied and fundamental research. He has been involved in the organization of conferences, workshops and tutorials to discuss and disseminate human factors knowledge.

CHARLES VAN DER MAST has a PhD in Computer Science from Delft University of Technology. He is associate professor at the Man-Machine Interaction group at Delft University of Technology. He teaches courses on Multimodal Interfaces and Virtual Reality, Developing Highly Interactive Systems, Multimedia, Educational Software and Intelligent User Experience Engineering. He co-developed the Bachelor/Master curriculum in Media & Knowledge Engineering. His interests include using various media to improve teaching, VR therapy for phobia treatment and agent support in complex systems.

NICOLA CAPPIELLO has a Master degree in Telecommunication Engineering. He is currently working at UL International as Handler Engineer. During his master thesis, he worked at Delft University of Technology on telemedicine architecture and on and evaluated a virtual personal assistant for telediabetology applications. He graduated at Politecnico di Milano and did a one year internship at Lund University.