MODELLING AND FORMAL SPECIFICATION OF A MULTIAGENT TELEMEDICINE SYSTEM FOR DIABETES CARE

Iñaki Martinez-Sarriegui[1,2], Hong Zhu[3], Lijun Shan[4], Gema García-Sáez[1,2], Enrique J. Gómez[1,2], M. Elena Hernando[1,2]

[1] Bioengineering and Telemedicine Centre, Politechnical University of Madrid, Spain
[2] CIBER-BBN Networking Research Centre, Spain
[3] Department of Computing, Oxford Brookes University, United Kingdom
[4] Department of Computer Science, National University of Defence Technology, China
imartinez@gbt.tfo.upm.es, hzhu@brookes.ac.uk, lijunshancn@yahoo.com, ggarcia@gbt.tfo.upm.es, egomez@gbt.tfo.upm.es, elena@gbt.tfo.upm.es

Keywords: Software agents, MAS Formal Specification, Modelling, Diabetes Care, Telemonitoring, Telemedicine

Abstract: This paper presents the modelling and formal specification of a telemedicine system for diabetes care. In such scenario, the multiagent technology supports the distributed autonomy of several Personal Assistants; the communications between them and the hospital´s agents; the control of the system´s access and multitask functionality; scalability; adaptability; robustness; and the provision to the physicians with the necessary automatic processing tools for the analysis of the large amounts of data generated by patients. We evaluated the AOIS meta-model and the CAMLE's modelling environment concluding that this methodology is adequate to represent a complex medical system like the one presented. The model and the formal specification provide a more complete view of the system and contain very useful information to cope with the future system evolution.

1 INTRODUCTION

Diabetes is the fourth leading cause of global death by disease. Currently 246 million people worldwide suffer from diabetes and the forecast for 2025 is that it will increase to affect 380 million (IDF, 2003). Diabetes Mellitus is a chronic disease characterised by a sustained elevated blood glucose level, caused by a reduction in the action of insulin secretion where related metabolic disturbances generate severe, acute and long-term complications that are responsible for premature death and disability (De Leiva et al., 1995). Effective control of patients' blood glucose level minimises the progression of the disease and reduces the risk of long-term neurological, renal and cardiovascular complications.

The treatment of diabetic patients attempts to achieve normal levels of blood glucose by maintaining a careful balance between diet, physical exercise and insulin injections therapy. Patients monitor their own blood glucose levels to take decisions regarding the adjustment of changes of insulin doses, meals and physical activity.

Due to its multifactorial and systemic character, Diabetes Mellitus has been considered a paradigm of chronic disorders which has led to an extensive application of information technologies. Nowadays telemedicine provides an integrated approach to information technology tools, which enhances co-operation between users, information and knowledge sharing.

Any telemedicine system for chronic care has to help patients' and physicians' decision making but supposes a huge quantity of data and increases users' workload.

To define and develop a telemedicine system for chronic care it is important to know well the characteristics of health care and then choose the best design and technology for the implementation. Health care is characterized by requiring distributed knowledge to solve the different problems requiring cooperation between professionals of different independent entities. Generally the problems are too complex and are decomposed in sub-problems easier to solve. It is needed a continuous access to medical information and this information has to be personal and proactive (professionals usually ask information about diseases or surgery before they have to face the problem).

The properties and features of multiagent systems (MAS) perfectly fit with the characteristics of health care above commented: distributed systems, sociability, management of distributed information, autonomy, proactivity, communication and coordination between separate entities (Nealon and Moreno, 2003).

There is a growing interest in the application of agent-based techniques (Wooldridge, 2002) to problems in the medical domain in different fields and oriented to several diseases. Some examples are Health Care at home (Koutkias et al., 2005), Health Care coordination (Aldea et al., 2001).In diabetes care there have been also some applications of the agents approach. In monitoring field we can see the SuperAssist project (De Haan, 2005), or the M2DM project (Hernando et al., 2003) and in management field the project proposed by Zhang et al. (2008)

Modelling plays a crucial role in the development of medical systems and its evolution as the main tool of requirements analysis and system/component design, representing the users' requirements with a set of agents at various granularities and organizing the agents into an information system.

This paper presents the adequacy of the AOIS meta-model (Zhu, 2005) for the modelling and formal specification of a complex telemedicine system for diabetes care (Hernando et al., 2004; Rigla et al., 2007).

2 MATERIAL AND METHODS

2.1 Modelling and formal specification

In the AOIS meta-model, the basic unit that forms an information system is the agent (Zhu, 2005). Agents are defined as real-time active computational entities that encapsulate data, operations and behaviours, and situate them in their designated environments.Agents perceive the visible actions and states of the agents in its environment and take actions and change state according to the situation in the environment and its internal state. A multiagent system (MAS) consists of a group of agents.

In the conceptual model, the classifier of agents is called caste, the basic brick of MAS. Caste serves as a template that describes the structure and properties of agents. Agents are instances of castes. Caste allows dynamic classification. That is, an agent can change its caste membership (casteship) at run-time. An agent can take an action to join a caste or retreat from a caste at run-time. When an agent joins/retreats from a caste, it will obtain/lose the structural and behavioural features of the caste. Dynamic casteship allows users to model the real world with MAS naturally and to maximize the flexibility and power of agent technology.

In the AOIS model, state variables and actions of an agent can be visible or invisible (internal). Agents communicate with each other by taking visible actions and changing visible state variables, and by observing other agents' visible actions or statesVisible actions are only observed by those agents interested in the agent's behaviour.

Agents in a MAS are designed and implemented with a designated environment. In other words, the environment of an agent is specified but allowed to vary within a certain range when an agent is designed.

2.1.1 CAMLE

The *Caste-centric Agent-oriented Modelling Language and Environment* (CAMLE) is a language that employs the multiple views principle to model complicated systems (Shan et al., 2006). In CAMLE there are three types of models that may consist of one or more diagrams:

- A *caste model* normally comprises one caste diagram with a set of caste nodes representing various types of agents in the system, and a set of links representing various relationships between agents of the castes.
- A *collaboration model* may consists of a set of scenario-specific collaboration diagrams that represent the interactions between agents in specific scenarios, and a general collaboration diagram that summarises the communications between agents.
- A *behaviour model.* It describes an agent's behaviour in terms of how it acts in certain environment scenarios at the micro-level.

2.1.2 SLABS

The *Specification Language for Agent-Based Systems* (SLABS) bridges the gap between graphic modelling and implementation in the AOIS development process (Zhu 2001, Zhu 2003).

The manual production of a multi-agent systems formal specification is labour-intensive, costly, time consuming and error-prone. The CAMLE modelling environment automatically generates the formal specifications from the graphic models.

2.2 Modelled system

The following scenario illustrates the functionalities of the telemedicine system covered in the modelling and specification.

Scenario 1:

Carmen is a type 1 diabetes patient that is followed up with a telemedicine service. She is provided with a PDA running a special application for diabetes management that download monitoring data from glucometers and insulin pumps using wireless communication. Additionally she can insert her blood glucose levels manually, view graphics and review all her monitoring data that are stored in a light database on the PDA. She has to periodically synchronize the PDA database with the one in the hospital. She could use the telemedicine Web application to insert the data directly to the hospital database but she prefers the PDA because she feels more autonomy with it.

Scenario 2:

Luisa is an endocrinology physician. She consults patients' monitoring data through the Web. She selects from which patient she wants to view data, having the opportunity of viewing graphics that helps her to take therapeutic decisions. According to the data received from patients, she decides to change their insulin therapy. Therapies are stored in the hospital database and patients receive the new therapy after synchronization.

Scenario 3:

Carmen synchronizes the databases again and receives a message informing she has a new therapy. She views the new therapy and configures the insulin pump with the new therapy's data. A reminder tool is set to warn the patient if the synchronization period is longer than a pre-fixed number of days.

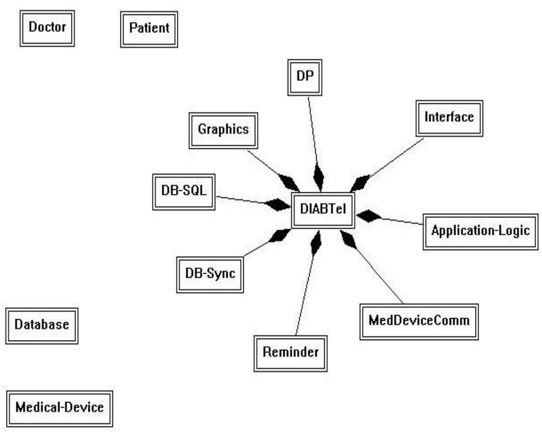

Figure 1: Caste Diagram.

3 RESULTS

The modelling of the telemedicine system has been performed by specifying caste diagram, collaboration diagrams and behaviour diagrams.

3.1 Caste Diagram

We identified two different types of castes: **actors** and **software agents** in the telemedicine system. Figure 1 presents the caste diagram.

3.1.1 Actors

Actors are the *users* (patients and doctors), the *medical devices* they use (insulin pump and glucometer) and the *databases* of the system. Actors are not agents but have to be represented in the model in order to clarify the working process of the system.

3.1.2 Software agents

We define 8 types of software agents as illustrated in Figure 1:

- *Interface*: This agent represents the graphic interface of the applications. It allows users to access the functionalities of the system.
- *Application-Logic*: It is the logic of the users' application and represents the applications intelligence. It knows all the software agents of the system and communicates with them to provide the functionality demanded by the user.
- *Reminder*: It warns the patient after *a pre-fixed number of* days since the last synchronization.
- *DP*: This agent is in charge of automatic data pre-processing when new monitored data is received from the PDA.
- *Graphics*: It is the responsible of the on-line generation of graphics on users' demand.
- *MedDeviceComm*: This agent manages the communication between the applications and the medical devices
- *DB-SQL*: This agent is the only one that can access the system data and manages and filters the operations of other agents with the database.
- *DB_Sync*: This agent manages the process of database synchronization.

3.2 Collaboration diagrams

The main collaboration diagram describes all the communications between agents. Additionally, we created eleven specific communication diagrams - one for each of the different functionalities offered by the system in the scenario described in section 2.2 - detailing not only the communications but also the sequence of them.

We classify the eleven specific communication diagrams into three types according to participants: patient, doctor or both:

- ‘Patient Diagrams’: *Configuration of Medical Device*; *Downloading of data from Medical Device*; *Database Synchronization*; *Database Synchronization Reminder*.
- ‘Doctor Diagrams’: *Patient Selection*; *Creation/Modification of a New Insulin Therapy*.
- ‘Patient/Doctor Diagrams’: *Viewing of Data*; *Insertion of Data*; *Viewing Graphics*; *Login on the system*; *Viewing Therapy*.

Figure 2 shows an example of the collaboration diagram for database synchronization. The patient can use an application running on a PDA (Interface and Application-Logic) and store the data on a local database, a light version of the one used in the hospital. The patient has to synchronize both databases. When he/she executes this action the Application-Logic asks the patient’s password to check the identity. In this process the Application-Logic consults the database to verify the password.

Once the patient has been identified the Application-Logic starts the synchronization process. For security reasons, the access to databases is restricted and has to be made through DB-Sync agent, which is responsible of talking with the two databases to really start the process. When the databases have exchanged the data, DB-Sync agent informs Application-Logic with the result of the synchronization.

After the synchronization process, the first Application-Logic action is to check the database for new therapy prescription or messages. If there is any new therapy, the Application-Logic agent informs the patient. After that it communicates to DP agent that a synchronization process has been made and stores the event on the database. All the events in the system are stored for auditory reasons.

The data downloaded from the insulin pump are stored in the PDA database in a compressed mode due to their size. In hospital database there are no restrictions of size like in the PDA so those data can be uncompressed. This task is made by DP agent accessing the hospital database, processing the data and storing them again.

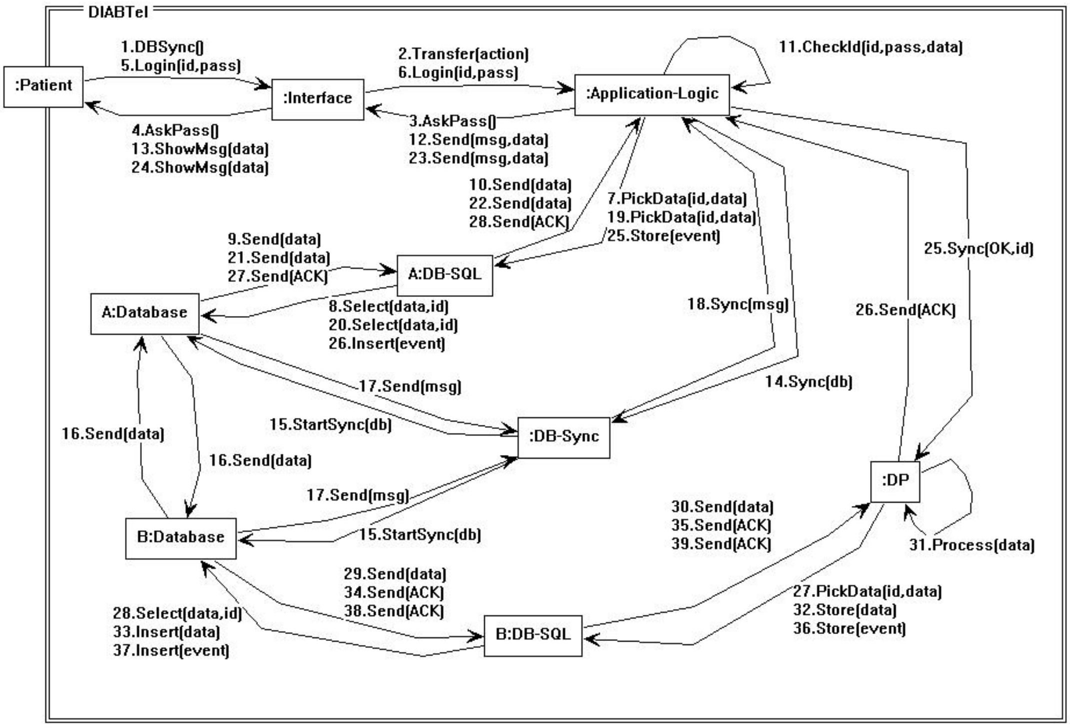

Figure 2: Collaboration diagram of Database Synchronization functionality.

3.3 Behaviour diagrams

In the behaviour diagrams we detail the behaviour of the agents of the system by specifying the rules that need to be followed to provide the desired functionality.

In the collaboration diagrams we can see the sequence of messages, while on behaviour diagrams this sequence is illustrated giving the conditions that have to be accomplished for the specific sequence. For instance, “one database cannot send data to other database if it does not receive the order from DB-Sync agent” and “the DB-Sync agent cannot order the start of a synchronization before the Application-Logic specifies it”.

The behaviour diagrams of some of the agents/castes are very large because they developed a lot of functions in the system. In Figure 3 we illustrate the behaviour diagrams of Graphics caste and Reminder caste as an example of the behaviour diagrams of the system.

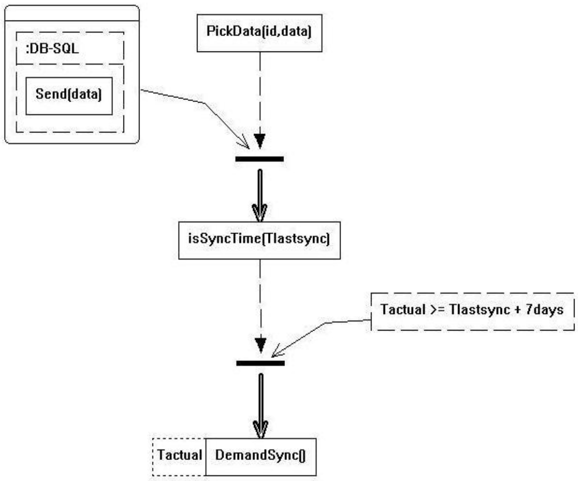

Figure 3: Behaviour diagram for: Reminder caste

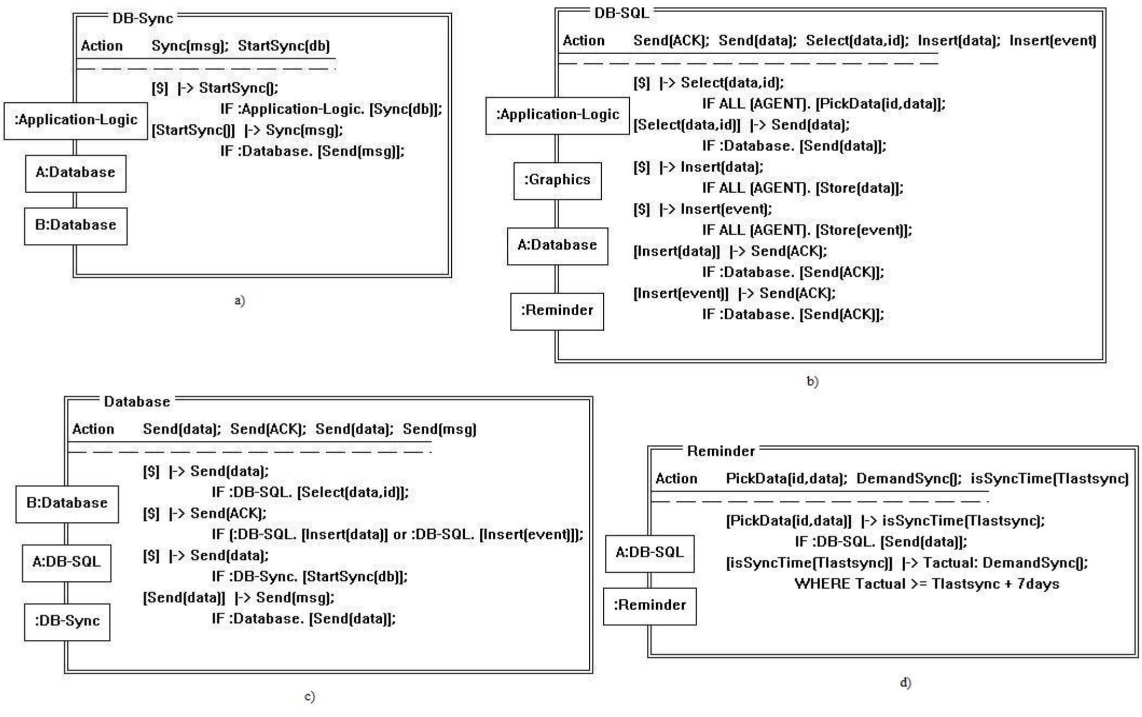

Figure 4: Formal Specification diagrams of the castes: a) DB-Sync, b) DB-SQL, c) Database and d) Reminder

3.4 Formal Specification

Once the model of the system is constructed, the formal specification can be automatically generated with CAMLE tool. Figure 4 shows an example of the diagrams of the DB-Sync caste, DB-SQL caste, Database caste and Reminder caste. When the definition of the model and the formal specification are finished, the implementation phase can start and can be easily faced by the developers.

4 DISCUSSION

The described system supposes a step forward to the goal of an 'ambulatory artificial pancreas'. The evolution of medical devices is creating a very promising situation. However, its ambulatory use requires the integration of close loop algorithms and medical devices in Personal Assistants running on portable terminals with communication capacities, providing patients with mobility possibilities, decision support tools and doctors' remote supervision and, at the same time, autonomy on their decisions.

An 'ambulatory artificial pancreas' is a complex concept. In such scenario, the multiagent technology supports the distributed autonomy of several Personal Assistants; the communications between them and the hospital´s agents; the control of the system´s access and multitask functionality; scalability; adaptability; robustness; and the provision to the physicians with the necessary automatic processing tools for the analysis of the large amounts of data generated by patients.

We can conclude that the multiagent approach is the approximation that better fits with those needs and the AOIS meta-model facilitates the definition and design of the required architecture. The graphical interface of the CAMLE's modelling environment tool helps in the process of modelling, allowing an easy creation of the different model diagrams and providing an automatic checking of the model consistency. The CAMLE tool also allows the generation of the formal specification directly from the different diagrams, which is a unique feature among other similar environments where the formal specification has to be manually done.

The presented model describes the current functionality of a working multiagent system that started to run in a hospital six years ago. The model and the formal specification provide a complete view of the system and contain very useful information to cope with the future system evolution.

REFERENCES

Aldea, A. et al. 2001. A Multi-agent System for Organ Transplant Co-ordination. *In the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, LNCS 2101. Springer-Verlag.

De Leiva, A., Lefèbvre P. and Nerup J., 1995. European dimension of Diabetes Research. *Diabetologia*, 39, p. 5-11.

De Haan, G., O.B. Henkemans, and A. Aluwalia. 2005. Personal Assistants for Healthcare Treatment at Home. *In the 2005 annual conference on European association of cognitive ergonomics*. Ghania, Greece.

Hernando, M.E., et al. 2003. Multi-agent Architecture for the Provision of Intelligent Telemedicine Services in Diabetes Management. *In the workshop on Intelligent and Adaptive Systems in Medicine*. Prague, Czech.

Hernando, M.E, García-Sáez, G., Gómez, E.J. and del Pozo, F. 2004. Intelligent Alarms Integrated in a Multi-Agent Architecture for Diabetes Management. *Transactions of the Institute of Meassurement & Control*. 26(3), p 185-200.

International Diabetes Federation, 2003. *Diabetes Atlas*, Brussels, 2nd Edition.

Koutkias, V. et al. 2002. A Multiagent System Enhancing Home-Care Health Services for Chronic Disease Management. *IEEE Transactions on Information Technology in Biomedicine*, 9(4)

Nealon, J.L. and Moreno, A., 2003. Agent-Based Applications in Health Care. In *Applications of Software Agent Technology in the Health Care Domain*. Basel, Germany. pp. 3-18.

Rigla et al. 2007. A Telemedicine System that Includes a Personal Assistant Improves Glycemic Control in Pump Treated Patients with Type 1 Diabetes. *Journal of Diabetes Science and Technology*. 1(4), p. 505-510

Shan, L., Shen, R., Wang, J., Zhu, H., 2006. Caste-centric Development of Agent Oriented Information Systems, *Handbook of Research on Nature Inspired Computing for Economy and Management*, Jean-Philippe Rennard (Ed.), Idea Group Inc. p. 692-707.

Wooldridge, M. 2002. An introduction to Multiagent systems. John Wiley, Chichester.

Zhang et al. 2008. Enhance Collaboration in Diabetic Healthcare for Children using Multi-agent Systems. AT2AI-6 Working Notes, *From Agent Theory to Agent Implementation*, AAMAS 2008, Estoril, Portugal, EU.

Zhu, H., 2001. SLABS: A Formal Specification Language for Agent-Based Systems, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 11. No. 5, pp. 529~558.

Zhu, H., 2003. A formal specification language for agent-oriented software engineering, in Proc. of AAMAS'2003, Melbourne, Australia, pp 1174 – 1175.

Zhu, H. 2005. Formal Reasoning about emergent behaviour in MAS, Proceedings of SEKE'05, Taipei, pp 280-285.