Medical protocols for taking decisions, in an ubiquitous computation context

Eladio Domínguez, Beatriz Pérez, Áurea Rodríguez, María A. Zapata

Dpto. de Informática e Ingeniería de Sistemas,
Facultad de Ciencias. Edificio de Matemáticas.
Universidad de Zaragoza. 50009 Zaragoza. Spain.
{noesis,beaperez,arv852,mazapata}@unizar.es

Abstract. Ubiquitous computation refers to the creation of environments with high processing and communication capacities in which the computers practically go unnoticed by the users. A context in which this vision of the future opens different lines of research is the field of health. The goal of this paper is to present a proposal for ubiquitous computation in a hospital context for the application of medical protocols that help with the taking of decisions. As a case study we use a protocol of action for the prevention, diagnosis and treatment of the infections caused by central catheters.

1 Introduction

The term ubiquitous computation, as coined by Weiser [17], refers to the creation of environments with high processing and communication capacities in which the computers practically go unnoticed by the users [16].

One of the promises of this vision of the future is that, in so far as the computers are capable of gathering and storing information from the environment, it will be able to better help the user to reach his goals [11]. The aim is that the interaction with the user adapts itself to the particular needs of the user and to his previous behaviour, being able even to anticipate his (her) actions.

The goal of this paper is to present a proposal for ubiquitous computation in a health context for the application of medical protocols that help with the taking of decisions. The concrete case of protocol that we use as a case study is a protocol that the sanitary personnel must follow in the Intensive Care Unit (ICU) of a hospital when they suspect that a patient has an infection caused by an intravenous catheter (IRC).

In the following section we explain the scenario that we try to develop once the whole system had been implemented. In that section we intend to clarify our goals. In the following section we present particular aspects of our concrete proposal. We will start by describing the architecture hardware that we propose to be able to go on to analyze and to present, later on, how we tackle the implementation of the medical protocols and the interaction between the user and the system. Finally, the current state of the project is explained.

2 The scenario

A patient comes to the ICU (Intensive Care Unit) from another service of the hospital with a radiofrequency identification card (RFID RadioFrecuency IDentification) [3, 9]. The clinical history number, identifying the patient in an unique way, his (her) clinical conditions, symptoms, diagnosis and the service which sends the patient to the ICU have been previously stored on the microchip of the card. On entry to the ICU there is a card reader, which sends a set of radio waves to the card, with the result that the card catches them by means of a small antenna. These waves activate the microchip, which, by means of the microantenna and the radio frequency, transmits the information that it stores. At the same time, the antenna sends the information to the ICU's server, which has software that is responsible for storing the information received in the Departmental Database, registering furthermore, the exact time when the patient is admitted. When this entry is registered in the database, the clinical history of the patient is requested, in an automatic way, from the Central Database. The clinical history is stored in the ICU's database in such a way that users can only read but not modify the information.

Once the patient is admitted, a catheter is inserted and, later on, a possible infection caused by the catheter is observed. In this situation, the doctor, after his (her) authentication and the identification of the patient by means of a PDA [18], gains access to the set of symptoms established by the protocol on catheter infections. The doctor indicates the group of symptoms observed in the patient and, in turn, the application communicates to him (her) the possible treatments and tests established by the protocol, which results possibly, in a change of the patient's state. The doctor chooses, if necessary, the treatment and the tests that he (she) considers to be more appropriate. This information is transferred, by means of WiFi or Bluetooth technology, to the Departmental database server.

The doctor finishes his (her) shift, and he (she) is worried about the state of the patient, he (she) sends a message to the computer system requesting to be informed, by means of SMS messages, of any changes in the state of the patient.

When the situation of the patient gets worse, the system automatically sends to the doctor's mobile the corresponding SMS messages.

Faced with the critical state that the patient is in, the doctor visits a work partner in order to analyze the situation. In the office of the latter, both colleagues accede, through the intranet of the hospital, with the correct authentication and the identification of the patient, to his (her) evolution in the ICU and, if it is necessary, to his (her) clinical history.

Finally, the patient reacts favourably to the treatment and the infection disappears, which is registered in the Departmental database server. All the information regarding the states, treatments and tests carried out, is stored in the Departmental database, as part of the local clinical history of the patient.

When the patient is sent to another service, his (her) transfers are registered by the card reader. The registration causes the transfer of the clinical information generated in the ICU and the deletion of the clinical history that was sent by the server at the time of the patient's admission.

3 Hardware architecture

With regard to the described situation, which is the declared goal of our project, we set out the proposed hardware architecture in figure 1 which shows all the hardware elements and the mobile devices that will make up the system.

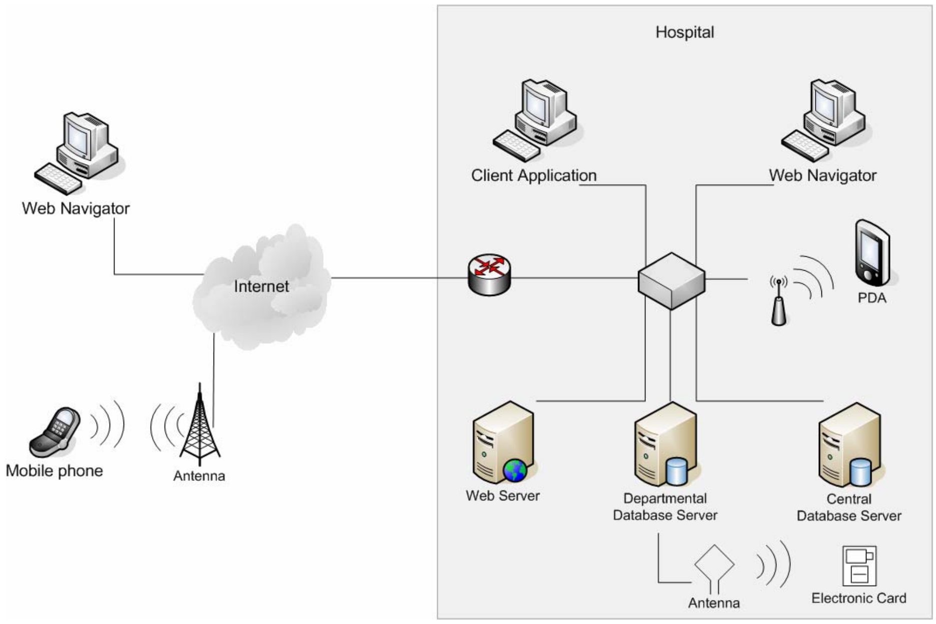

Fig. 1. Hardware architecture.

It is supposed that the hospital has a central server with two databases, one that contains the information about patients with his (her) clinical histories and another one that contains the information about the clinical personnel. The hospital also has a Web applications server, by means of which an intranet application is managed, which is addressed to the clinical service. On the other hand, the ICU has a local server for the management of its own information. Just as has been mentioned above, when a patient is admitted in the ICU, the Departmental Database requests the clinical history of the patient from the Central Database, which is stored as an unmodifiable file. Thus avoiding the highly undesirable situation that the doctor cannot consult the clinical history of the patient in a critical state because of a breakdown in the net.

In general, all databases, irrespective of their location, collaborate through a federated system architecture interchanging information through services requests [4]. The goal of the project is to include mobile elements in the system and to construct the applications that are necessary in order that the described scenario can be real.

4 Medical protocols

Initially the protocols are on paper, at the disposal of the clinical personnel. The protocols are hospital dependent so that they must be freely accessible especially to new personnel.

Moreover, the medical protocols, due to their individualized characteristics, must be modified with the passage of time in order to adapt both to the evolution of medical knowledge and to the advances that can take place in pharmacological treatments or in clinical tests. For this reason, at the moment of deciding the way in which the protocols are going to be implemented, it is important to take into account in our decision whether the resultant code and data structures are easy to be maintained or not. A medical protocol establishes the interventions that must be carried out with regard to the current circumstances of a patient. Taking this into account, we consider that there are two types of changes which can be produced in a protocol:

- modification of static aspects of the protocol: for example the symptoms that must be considered, the treatments to be given, the tests to be taken...
- modification of dynamic aspects of the protocol: for example the moment at which it is necessary to consider a symptom, when a treatment begins, or at what moment a clinical test is to be requested...

The protocol that we are using for the development of the project is a protocol that the sanitary personnel must follow in the ICU when it is suspected that a patient has an infection related to intravenous catheter (IRC). The use of vascular catheters for diagnostic or therapeutic purposes is more and more frequent, especially in patients in a critical state or with severe pathologies or serious chronic illness. The problem is that the infections caused by catheters constitute the principal cause of hospital-acquired nosocomial bacteremia and are related to a high morbidity and high mortality rates [2, 12]. On account of all this intervention protocols for the prevention, diagnosis and treatment of infections caused to central catheters must be established.

In particular, an action protocol establishes the criterion that the doctor must take into account to decide whether the catheter can be kept in or if, to the contrary, it must be removed and to decide what antibiotic treatment must be given to the patient. The range of possibilities that the protocol permits depends principally on the clinical characteristics of the patient, on the concrete symptoms that the doctor observes and on the results of the clinical tests. Thus, the protocol helps the doctor to decide whether to remove the catheter or not and what antibiotic treatment is to be given, with regard to the patient's condition at every moment.

4.1 Adaptation to static changes

In order to adapt the computer tool when static aspects of the protocol are modified, we have considered it appropriate to store all the static information in a database, which we will call *protocol database*. This database will contain the symptoms, clinical characteristics, pharmacological treatments and clinical tests that are related to the application of the protocol. A change of any one of those aspects of the protocol will suppose only the modification of the information stored in the database, without, in principle, it being necessary to modify the code of the programs.

4.2 Adaptation to dynamic changes

The achievement of a completely satisfactory solution for the case in which the changes take place in the dynamics of the protocol is more difficult. The explanation of our approach is the following.

Several formalisms have been proposed in the literature for the specification of systems behaviour. The modelization of diagnostic and therapeutic processes has been tackled using both formal and informal methods [1].

The problem of informal methods is their ambiguity and that the models can not be formally analyzed. On the other hand, formal methods do not have the problems of informal methods but, in general, they are more difficult to understand [5, 6, 7] (specially for people such as doctor who typically do not have a specific mathematical background). In [1] it is claimed that the modelization of medical processes must be done by means of methods that overcome the problems of formal and informal approaches. Methods that, on the one hand, have an easily understandable notation and, on the other hand, can be formally analyzed.

Following this criterion we have considered appropiate to use the statechart technique [8] in order to represent the dynamics of the protocols. We have taken this decision in view of the fact that several proposals value positively the notational facility of the statecharts to make it easily to understand the specific case of the representation of diagnostic and therapeutic processes [10]. All this with the advantage that this easily understandable visual formalism is at the same time formal and rigorous. It must be acknowledged that, to our knowledge, nobody has used the statecharts for the specific case of the representation of medical protocols. For this reason our approach would open a new field for the application of these statecharts.

In our particular case the states of a statechart can represent what the different states of a patient can be, as the protocol is applied to him (her). The transition between two states will represent the events that cause patient to change state, for example, as a consequence of results obtained from the clinical tests, or because the doctor considers it appropriate.

Once the statechart that represents a protocol has been established, we make double use of it. On the one hand, following a design methodology controlled by models, the statechart will help us to construct the program that implements the protocol. And on the other hand, it will help us to fix the structure of the database in which the clinical history of the patients will be stored indicating the different states through which they have passed during the application of the protocol.

There are a variety of proposals in the literature about how to implement a statechart (see, for example, [13, 14, 15]). In general two types of approach can be distinguished:

- One can be defined as a global variable whose purpose is to store the current state and the action to be taken is decided by means of conditional sentences with regard to the value of that variable and the event that has happened.
- Every state is implemented by means of a class and every event is implemented as a method of the origin class of the transition that the event triggers.

In our particular case we consider more appropriate the second option since, though the first one needs less space for storage, the second one has the advantage that it generates code that is easier to maintain when changes are produced [15].

However, although we have used the ideas of the authors that follow the second option as a starting point, our proposal has several characteristics that make it different. In general, the authors are only concerned about constructing a program whose code determines the behaviour of an object which lives and dies during the execution of the program. In our case the needs are different. On the one hand, the functionality that we want for the PDA needs that the application starts by requesting from the database the current state of the patient and that, after the decision of the doctor, if it is the case, it stores the new state of the patient. In this way our solution has to equate the implementation of the statechart with the management of the objects persistence (something that other authors do not address). And on the other hand, besides the implementation of the dynamics represented in the statechart, we also want to use the statechart as a guide which helps how to store the patient states, in order to obtain the backward traceability of the patients evolution with regard to the application of the protocol. For this reason our proposal also includes the definition of a database schema for the storage of the information related to the patients states.

5 State of the project

The databases that store the information with regard to the patients, the sanitary personnel, the protocol and the patients states, have been implemented using the agent database Oracle 10g. Since the Oracle Application Server 10g is optimized for ORACLE Database 10g, we have considered it appropriate to use it as an applications server.

The application of the Web server that shows information with regard to the application of the protocol is being developed with the JSP. Regarding the client application that will allow the management of the protocol and, particularly, the maintenance of the information concerning the medical tests, medication and other elements that define the protocol; this is implemented in Visual Basic. The PDA application that allows the doctor to consult in real time information concerning the protocol, is implemented with technology .NET.

Finally, it is to be noted that in order to achieve communication between the mobile and the Web server by means of SMS messages, different options have been considered. Possibly the best option will be to use a service provided by a mobile telephone company that allows the above mentioned requests to be performed by an existing application.

Among all the elements that take part in the scenario we have suggested, we consider that the PDA application is the one that is most innovative, both from the technological point of view due to the technical characteristics of the device, as well as from the stand point of the difficulty of the particular form in which we implement the statecharts, such as has been previously explained.

Acknowledgement.
We are especially grateful to Dr. Francisco Olmos for the help that he has provided in order to understand and acquire the necessary knowledge about the protocol of intervention in view of infection with respect to infections caused by catheter, that we have used. Also we would like to thank the university tutor Ángel de Miguel for his close collaboration on the project and the university tutors Jorge Lloret and Ángel Luis Rubio for their help in the development of the paper.

Besides the authors of the present paper, we would like to thank Maribel Honrubia, Jéssica Navarro and Maore Velaz, who are also working on the development of this application.

References

1. L. Baresi, F. Consorti, M. di Paola, A. Gargiulo, M. Pezzè, LEMMA: A Language for an Easy Medical Model Analysis, *Journal of Medical Systems*, Plenum Publishing Co. 21 (6), 1997, 369-388.
2. E. Bouza, J. Liñares, Á. Pascual, Diagnóstico microbiológico de las infecciones asociadas a catéteres intravasculares, capítulo 15, A. Pascual (coord.), en *Procedimientos en Microbiología Clínica, Segunda edición*, E. Cercenado, R. Cantón (eds.), Sociedad Española de Enfermedades Infecciosas y Microbiología Clínica, 2004, http://www.seimc.org/protocolos/microbiologia/, consultado: abril, 2005.
3. S. d'Hont, *The Cutting Edge of RFID Technology and Applications for Manufacturing and Distribution*, http://www.ti.com/tiris/default.htm, consultado: abril, 2005.
4. E. Domínguez, N. Álvarez-Gil, L.A. Barcelona, J.C. Ciria, A. De Miguel, A. Francés, N. González, J. Lloret B., J. Lloret G.,A. Rodríguez, M. A. Zapata, Modelo de servicios: una perspectiva conceptual, *Actas de la Conferencia IADIS Ibero-Americana WWW/Internet 2004*, J. M. Gutiérrez, J. J. Martínez, P. Isaías (Eds.), IADIS Press, 2004, 203-210.
5. E. Domínguez, A. L. Rubio, M. A. Zapata, A Way of Dealing with Behaviour of State Machines, *Proceedings of the UML 2000 Workshop: Dynamic Behaviour in UML Models: Semantic Questions*, 2000, 32-37.
6. E. Domínguez, A. L. Rubio, M. A. Zapata, Meta-Modelling of Dynamic Aspects: The Noesis Approach, *Proceedings of the ECOOP'2000 International Workshop on Model Engineering*, 2000, 28-35.
7. E. Domínguez, A. L. Rubio, M. A. Zapata, Dynamic Semantics of UML State Machines: A Metamodeling Perspective, *Journal of Database Management*, 13 (4), 2002, 20-38.
8. D. Harel, Statecharts: A visual formalism for complex systems, *Science of Computer Programming*, 8, 1987, 231-274.
9. Information Highway group, *Tecnología RFID*, http://www.ihg.net/java/X?cgi=lateral.rfid.Introduccion.pattern, consultado: abril, 2005.

10. N. Kam, I.R. Cohen, D. Harel, The Immune System as a Reactive System Modeling. T Cell Activation with Statecharts, *Proc. Symposia on Human-Centric Computing Languages and Environments*, IEEE, 2001, 15-22.
11. M. Lease, Plan-Aware Behavioral Modeling, *Intel Research Seattle*, IRS-TR-02-013, 2002.
12. C. Leon, J Ariza, Guías para el tratamiento de las infecciones relacionadas con catéteres intravasculares de corta permanencia en adultos: conferencia de consenso SEIMC-SEMICYUC, *Enferm Infecc. Microbiol Clin.*, 2004, 22(2), 92-101.
13. P. Metz, J. O'Brien, W. Weber, Code Generation Concepts for Statechart Diagrams of the UML v1.1, *Object Technology Group (OTG) Conference*, University of Vienna, 1999.
14. I. A. Niaz, J. Tanaka, Code Generation from UML Statecharts, *Proceedings of the 7th IASTED International Conference on Software Engineering and Applications-SEA*, 2003, 315-321.
15. I. A. Niaz, J. Tanaka, Mapping UML Statecharts to Java code, *Proceedings of the IASTED International Conference on Software Engineering-SE*, 2004, 111-116.
16. M. Satyanarayanan, Pervasive Computing: Vision and Challenges, *IEEE Personal Communications*, 2001, 10-17.
17. M. Weiser, The Computer for the 21st Century, *Scientific American*, 265(3), 1991, 94-104.
18. A. Zarazaga, Las PDA en el entorno hospitalario. Mitos y realidades, *Informática y Salud*, 40, 2003, 16-22.