International Program Committee (IPC)

Honorary Chairs: Sgurev, V.
Boyanov, K.

Chairs: Batchkova, I. Boshnakov K.

Vice chair: Stoilova, K.
Vice chair: Aleksiev, K.
Vice chair for Industry: Andreev, V.

Members:

Atanasov, A	(BG)	Kocijan, J.	(SI)	Ruskov, P.	(BG)
Atanasov, K.	(BG)	Koprinkova, P.	(BG)	Simeonov, I.	(BG)
Dakovski, L.	(BG)	Liutov, N.	(BG)	Smrikarov, A.	(BG)
Damyanov, Ch.	(BG)	Madjarov, M.	(BG)	Spasov, G.	(BG)
Dotsinski, I.	(BG)	Marinov, E.	(BG)	Stoilov, T.	(BG)
Elenkov, G.	(BG)	Mihaylova, L.	(GB)	Stoyanov, S.	(BG)
Farhi, O.	(BG)	Mladenov, M.	(BG)	Tasheva, D.	(BG)
Filipov, F.	(BG)	Nenov, T.	(BG)	Tenekedjiev, K.	(BG)
Frey, G.	(GE)	Nikolov, E.	(BG)	Todorova, M.	(BG)
Garipov. E.	(BG)	Perev, K.	(BG)	Topalov, A.	(BG)
Georgieva, P.	(PT)	Petkov, P.	(BG)	Tzotchev, V.	(BG)
Grancharova, A.	(BG)	Petrov, M.	(BG)	Vachkov, G.	(FJ)
Hadjiiski, M.	(BG)	Popchev, I.	(BG)	Velev, K.	(BG)
Hristov, N.	(FR)	Puleva, T.	(BG)	Vesely, V.	(SK)
Ilarionov, R.	(BG)	Rujekov, G.	(BG)	Yordanova, S.	(BG)
Ishtev, K..	(BG)	Rumenin, Ch.	(BG)		

National Organizing Committee (NOC)

Chair: Hristova, N.

Vice co-chair: Gocheva, D.
Dimitrov, S.

Members:

Antonova, I.
Mishkov, R.
Ivanova, D.
Monova, E.
Nikolova, N.
Yonchev, A.
Dobrudjaliev, D.

Secretary: Dimitrova, E.

CONTENTS

Session A2.1: Process control (2)

Session A2.2: Intelligent control of power plant systems

Session A2.3: Measurement and analysis of clinical data

Session A2.4: Measurement and quality assessment

Session B1.1: Modeling and optimization of organization systems

Session B1.2: Analysis and control of organizational systems

Session B1.3: Software engineering (1)

International Conference
AUTOMATICS AND INFORMATICS'15
4-7 October 2015, Sofia, Bulgaria

MODEL DRIVEN DEVELOPMENT OF HOLTER MONITORING SYSTEM

Tz. Ivanova, I. Antonova, I. Batchkova

Dept. of Industrial Automation, University of Chemical Technology and Metallurgy, Bul. Kl. Ohridski 8, Sofia, Phone: +359 2 8163326 , E-mail: {iskra.antonova, idilia}@uctm.edu

Abstract: The main aim of the presented paper is to analyse the current model driven approaches based on UML for development of Ambient Intelligence places. The proposed approach is based on the combined use of MDA extended with the development of compositional models represented in two dimensions – domain and context. The suggested approach is illustrated with the development of Holter monitoring system that is an essential part of an ambient intelligence space in healthcare. Finally some conclusions are made.

Keywords: Model Driven Development, Model Driven Architecture, Ambient Intelligence, UML, Holter monitoring

1. INTRODUCTION

Ambient Intelligence (AmI) is a vision for smart, device based, sensitive environments that are reactive to people and able to make their actions safer, more efficient, more informed, more comfortable or simply more enticing. Different areas such as a computer networks, sensors, human-computer interfaces, ubiquitous computing, robotics and artificial intelligence support the development of so called AmI spaces. The problems and difficulties of achieving AmI may be summarized as follow: high cost of ownership, inflexibility, poor manageability and low level of interoperability.

Reliable way to overcome these difficulties and shortcomings is the choice of appropriate development approach that will ensure the achievement of reusability, portability, scalability and opportunities for rapid integration and a high degree of interoperability of devices and solutions constituting the AmI space. These requirements fully meet the approaches of Model Driven Development (MDD) [1], where the systems are presented as models that conform to meta-models, and the model transformations are used to manipulate the various representations. Model Driven Architecture (MDA) [2] is a remarkable MDD initiative of Object Management Group (OMG), consisting in transformation of different Platform Independent Models (PIM) towards executable applications. MDA is widely accepted to be a foundation for reusing domain knowledge and is treated as a domain engineering technique [3]. PIM represent conceptual design of functional requirements and are resistant against changes to the implementation technologies and software. It is intended to provide a comprehensive solution for application's interoperability and portability as it separates system design from concrete system architecture. In the core of MDA are the open standards - Unified Modelling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI), etc.

UML [4] does not specify a methodology for model driven software or system development but aims to provide an integrated modeling framework, covering structural, functional and behavior descriptions. An important aspect in the implementation of MDD for development of AmI spaces is covering the requirements for service-oriented applications as well as the peculiarities of AmI, which generally would be summarized as pervasive and ubiquitous computations, profiling and context-aware applications, etc. This requires concretization of MDD for the purpose of AmI, which is reflected in the development and use of special UML meta-models, profiles and tools.

The main goal of the paper is to present MDD approach for development of AmI spaces using context based UML profile based on an example from the field of health and medicine, as is the Holter monitoring system.

The paper is organized in 5 parts. After the introduction, in part 2 a short overview of Holter monitoring is presented. In part 3 some MDD approaches for development of AmI systems are discussed. The proposed MDD approach for development of Holter monitoring system is described in part 4. A case study concerning an application of the suggested approach is presented in part 5. Finally some conclusions are made.

2. OVERVIEW OF HOLTER MONITORING

Many heart problems become noticeable only during different type of everyday life activities and ambulatory electrocardiography (ECG) also known as Holter monitoring, ambulatory ECG or Ambulatory EKG has been used in order to support diagnostics and to prevent unpleasant incidents. For these purposes a patient wears a small recorder called a Holter monitor as he or she goes about normal daily life. Holter monitoring can be seen as a mobile expert system, which not only detects the presence of abnormal heartbeat intervals but also requests for context information about the activities undertaken by the patient upon detecting changed heart rate [5].

There are two basic types of ECG recording devices: continuous and intermittent recorders. Continuous recorders monitor about 100000 heartbeats in 24 hours and are likely to find any heart problems that happen with activity. Intermittent recorders are used over periods - weeks or months. During this time they provide brief, intermittent recordings. They are used when symptoms of an abnormal heart rhythm do not occur very often. An intermittent recorder can be used for a longer time than a continuous recorder. The information collected by an intermittent recorder can often be sent over the phone to a doctor's office, clinic, or hospital [6].

The main components of Holter monitoring system are shown in Fig.1 and include Holter with Bluetooth, personal cardio assistant (PCA) based prototype and expert system for ambulatory ECG beat detection. There is a possibility a smart phone to collect the information of received ECG signals. The so suggested architecture of Holter monitoring based on PDA (Personal Digital Assistant) must provide a high degree of mobility and appropriate user interaction to ensure preventive monitoring of heart activity while the patient is either in stationary or mobile situation. The system monitors cardiac rhythms from the ECG signal and upon discovering changes in heart rate will ask the user to enter information about its current activity of daily living.

The following sceneries are included in the monitoring system [7, 8]:

- In case of an alarm for small ECG changes, PEM displays a short message and invite the user to report about the message at the one of his next visits to his General Practitioner or cardiologist (Hospital). The message content and the automated interpretation results together with the ECG are stored in the memory card.
- In the case of a major alarm, the patient has the opportunity either to send of medium alarm or to authorize the sending of his/her ECGs and personal electronic Health record to the alarm server.
- Decision making embedded in PEM by detection of arrhythmias;
- Decision making embedded in PEM by diagnosis of ischemia;
- Generation of alarms according the severity of arrhythmia and/or ischemia and the additional clinical information;
- Management of alarm messages for handling any communication problems with the contacted health care providers – Emergency Call-center, General Practitioner, Hospital.

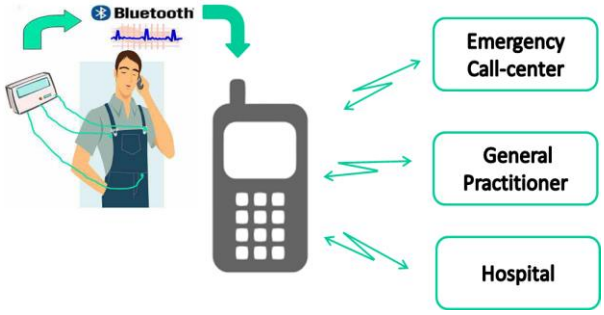

Fig.1: Proposed Holter monitoring system

3. OVERVIEW OF SOME APPROACHES FOR DEVELOPMENT OF AmI SYSTEMS

There are several different proposals for development of AmI systems and all of them are based on UML. The most popular are UML Profile for context modeling and ContextUML Profile, which will be discussed in short below.

3.1. UML Profile for Context modeling. The first large subclass of approaches uses context-aware application for AmI systems. A context-aware application for AmI can be modeled using specialized UML diagrams in order to separate concerns and to focus different aspects of the software system. Several works have been taken in the domain of context modeling. For this purpose a special UML Context Modeling Profile (CMP) was created [9]. This profile is a partial extension of a UML meta-model and provides a way to introduce a special terminology for a domain, to add semantics to meta-models and, which is vital, to restrict the way how to use the meta-model. It contains different stereotypes, tagged values and constraints assigned using OCL (Object Constraint Language) defining how an existing meta-class of the meta-model may be extended. The proposed meta-model is then used for modeling context [10]. New meta-classes must not be inserted in the class hierarchy of the meta-model because is forbidden to modify the class definitions in the meta-model. As modeling tool StarUML, which is open source modeling platform in the medical domain, is used. The authors aim to add new stereotypes in the feature in order to be able to model dynamic behavior of context-awere applications.

3.2. ContextUML profile. ContextUML approach is another UML-based graphical language profile aimed at the design of context-aware web services, which includes constructs for modeling context information. It provides constructs for generalizing context provisioning that includes context attributes specification and retrieval and formalizing context awareness mechanisms and their usage in CASs (Context-Aware Services). ContextUML distinguishes between atomic and composite context. However the notion of composite context is different from other approaches which also consider it. ContextUML defines a composite context as a state of a context depending on two or more atomic contexts, and models it by using statecharts [11]. ContextUML models can be expressed in MLContext which is also an MDD approach for the development of context-aware systems. The MLContext meta-model has been designed around the concept of entity. The context of use of a system is represented by defined set of elements that represent all contextual aspects. The type of a contextual element is provided by one of the following entities: the user, the application, the environment or the behavior. So each contextual element has one of the following stereotypes: “UserContext”, “ApplicationContext”, “EnvironmentContext” and “BehaviorContext”. The context information of a simple context can contain a reference to other entities of the model. The information from the source is supplied through a method (Method metaclass) from the physical device interface. The EMF framework of Eclipse as basic tool is used [12]. In ContextUML the acquisition of context information is not modelled. ContextUML covers the functionalities of context-aware services.

Modification or change in one element of the context of use involves the transition of this context toward another state of the user's current situation. In order to overcome this modification a specific UML notations and new elements - stereotypes, tagged values and constraints are added [13].

4. SHORT DESCRIPTION OF SUGGESTED APPROACH

The suggested model driven approach supporting the fulfillment of above mentioned requirements uses MDA and comprises the whole development life cycle of software development, starting with CIM meta-model to the deployment. It provides a powerful conceptual framework for development and transformation of three interconnected types of models - Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) towards executable applications as shown in Fig.2. The main features of the models are:

- CIM – is also known as business or domain model that uses vocabulary to present the basic expectation from the system and to bridge the gap between domain experts and developers. This abstraction hides all specifications connected to the system implementation.
- PIM – is a view of the system without any details about implementation. Basic tasks of the PIM model are to model logical data, to establish dependencies and defining workflows and processes. Furthermore, PIM models must be sufficiently complete and accurate to ensure a higher degree of automated implementation of the models in the next layer (PSM).
- PSM – combine the PIM specifications with concrete platform information needed for enabling system execution, i.e. the basic role of this model is to ease the code generation using PIM and selected execution platform.

Very important aspect of the MDA approach is the transformation between the different viewpoint models. Model transformation relies on a set of mapping rules between models that inform the transformer tool about the patterns it needs to follow to output the target model [14].

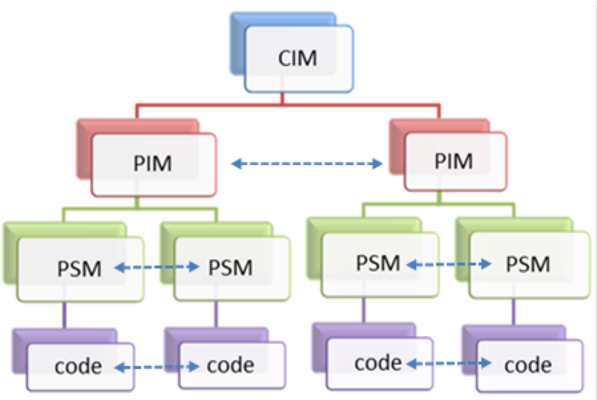

Fig.2: Illustration of the suggested MDD approach

An extension to the traditional MDD approach is the suggestion for use of compositional models as core elements for automation of expert activities, enabling the context-awareness of application. Compositional models encapsulate knowledge on a specific problem and are modeled in two dimensions – domain and context. Such a way the domain specific concepts have meaning only in relation to contextual concepts. Whereas the domain-specific models are formal descriptions of the domain concerned, the context-awareness models are representations of information related to the domain environment. As is evident from Fig.3 the domain-specific model uses measurements and diagnosis results based on measurements, while the context-awareness model is based on device and user parameters. The compositional model may be seen as a collection of data, rules, and terminology leading to transparent understanding of a particular process. The process implementation is supported by management, measuring and improvement procedures. The terminology of the compositional model allows describing user profiles, which may be aggregated in generic or reference profiles participating in quality assessments and process improvements.

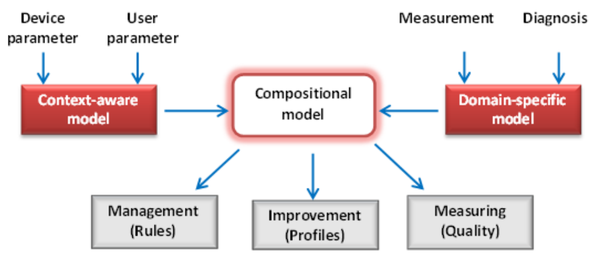

Fig.3: Use of compositional models

The suggested MDD approach is suitable for the development of AmI systems and its combination with SOA or MAS, because of the opportunities they create through the applied meta-models and model transformations for providing reusability, verification and validation of models, integration and interoperability.

5. UML BASED MODELS OF HOLTER MONITORING SYSTEM

The current research aims to demonstrate the suggested approach using UML diagrams representing a set of possible scenarios for a PCAs (Personal Cardio Assistant system), which is a typical example of systems for creating ambient intelligence [7].

Situations in which patients have a Holter monitoring system may be presented with set of scenarios. A possible scenario is modeled with UML use case diagram shown in Fig.4. This scenario includes PCA system and patient, a general practitioner, a hospital and an emergency center as actors. The Personal Cardio Assistant system should evaluate person's heart status and provide a user-specific recommendation. The scenario includes the follow use cases operated from patient – "CurrentState", in which "ArrhythmiaDetection", "AtrialFibrillation", "Tachycardia", "Barycardia" and "Normal", and "ECGRecording". The PCAs includes use cases named "ECGDiagnosis", "ECGRecording/Storin", where uses "WirelessTransmission" supporting "ECGStripSummary" and "BeckUp".

Depends on situation, the scenario can be modified and designed for users who have other cardiovascular problems. The implementation of any scenario aims to allow of the patient to become a specialized medical care in case of different cardiovascular problems when the patients leave the hospital or care centers.

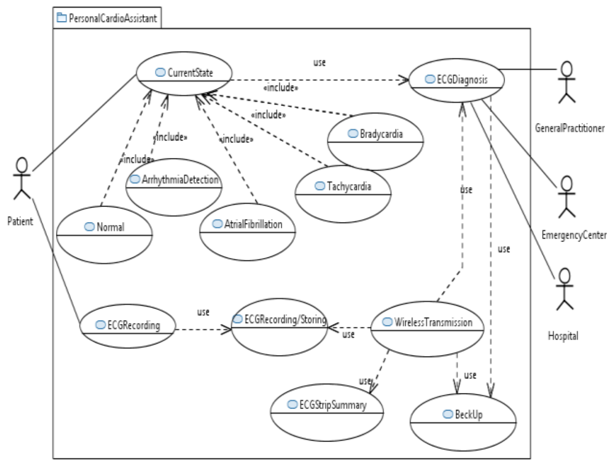

Fig.4: Use case diagram of personal cardio assistant

The functionality of PCA system is presented with class diagram shown in Fig.5. The PCAs is modeled as interface named "UserProfile". This interface includes the classes "Patient" and "HealtCareProfessional". Each "Patient" is connected with his "ClinicalSummary", "SensorSystem" and "Device". There is a possibility to make records of ECG, modeled as "ECGRecord" to each "ClinicalSummary".

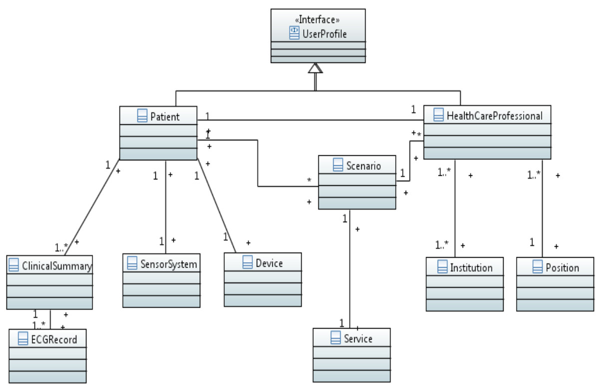

Fig.5: Class diagram of Holter monitoring system

The hardware structure of Holter, modelled as a class diagram and presented in Fig.6, consists of following classes:

“Electrodes”, “LCDMonitor”, “PreamplifierModule”, “SDModule”, “Microcontroler” and “Bus”.

In order to design of software modules of embedded system Real-Time operating system is used. The software of Holter monitoring system is separated in four modules – RT Analysis, Data transmission, Data storage and LCD display, modelled using UML state machine diagram. The system has three main tasks – data acquisition; data transmission, real-time analysis and storage, LCD control. Under different mode, different tasks will run.

The structures of software system and data transmission are presented respectively in Fig.7 and Fig.8. The software structure is separated in two parts, modelled as regions in State machine diagram – the first one represents the performance of microprocessor and the second one - the microcontroller. The system microprocessor’s become data and check if the cache is full. In case that the cache is full, the microprocessor interrupts the controller. Otherwise the microprocessor continues to get data.

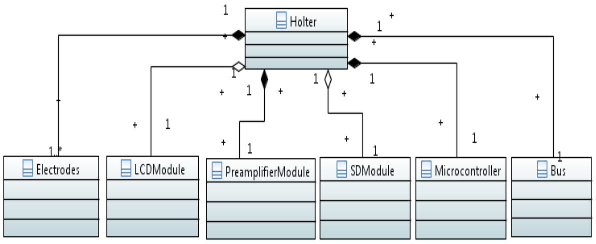

Fig.6: Hardware structure of Holter

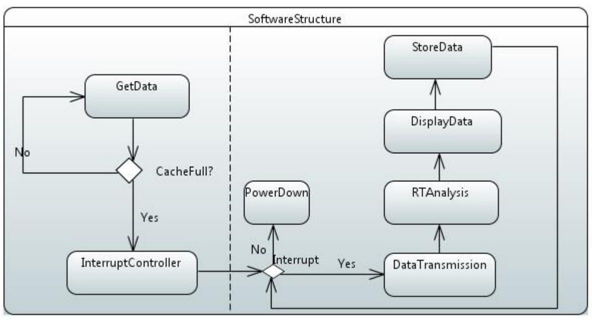

Fig.7: Structure of software system

The used microcontroller is based on CPU with low power consumption and has two low power modes – idle and power down. In the power-down mode, the oscillator closed, so there is no internal clock and power consumption down to almost zero. The states of processor are data transmission, RT analysis, display and store data. Data transmission (Fig.8) has two modes – normal and high speed mode. The normal mode is enough only for real-time ECG data transmission.

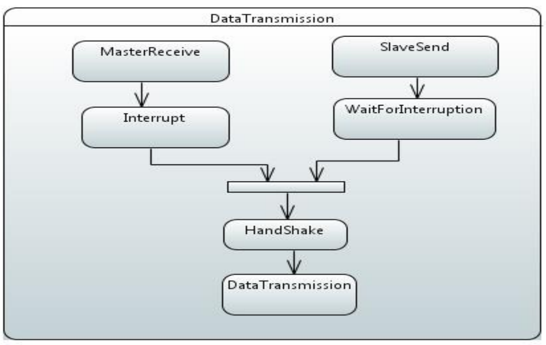

Fig.8: Data transmission

6. CONCLUSIONS

Ambient intelligence systems are a challenge in terms of both the intelligent methods and algorithms used and the developed software mainly because that the underlying idea is adapting the behavior of ambient intelligence system to the users, not vice versa. In connection to this, the development of ambient intelligence systems requires the use of new approaches in order to create the conditions to meet the requirements for these systems in terms of the following properties: non-obtrusive, context-aware, personalized, adaptive and anticipatory. The approach proposed in this paper attempts to cope with the requirement to achieve the context-aware of AmI applications using the model driven architecture and compositional models. An important next step in improving the approach is the use of an extension of the development environment using the profiles and metamodels for context UML.

REFERENCES

1. Kent, S., Model Driven Engineering. In Proceedings of IFM, 2002, LNCS 2335, Springer.

2. OMG-MDA. MDA Guide version 1.0.1. OMG document omg/2003-06-01, 2003.

3. Model Driven Architecture Home Page. OMG, 2011 [Online]. Available: http://www.omg.org/mda/ (Accessed: 31-Oct-2011).

4. OMG-UML, 2010. OMG. http://www.omg.org/.

5. Li J, A mobile ECG monitoring system with context collection, Master thesis, Dublin Institute of Technology, 2008.

6. Holter Devices Comparison, available on the website: www.medcompare.com/matrix/132/Holter-Monitor.html [Last accessed on 01/01/2009].

7. Krupaviciute A., Personalised wearable cardiac sensor services for pervasive self-care, PhD Thesis, INSA de Lyon, 2012.

8. Rubel P. at all, Toward personal eHealth in cardiology. Results from the EPI-MEDICS telemedicine project, Journal of Electrocardiology, Volume 38, Issue 4, Supplement, pp.100–106.

9. Simson Ch., CMP: A UML Context Modeling Profile for Mobile Distributed Systems, Proceedings of the 40th Hawaii International Conference on System Sciences – 2007.

10. Benselim M. S., Seridi-Bouchelaghem H., Extending UML Class Diagram Notation for the Development of Context-aware Applications, Journal of emerging technologies in web intelligence, Vol. 5, No. 1, February 2013.

11. Sheng Q. Z., Benatallah B, ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services. In 2005 International Conference on Mobile Business (ICMB 2005), 11-13 July 2005, Sydney, Australia. pp. 206–212. IEEE Computer Society, 2005.

12. Hoyos J.R., Garcia-Molina J., Botıa J. A., MLContext: A Context-Modeling Language for Context-Aware Systems, Proceedings of the Third International DisCoTec Workshop on Context-Aware Adaptation Mechanisms for Pervasive and Ubiquitous Services (CAMPUS 2010).

13. Benselim M. S., Seridi-Bouchelaghem H., Modeling context with extended UML, AWER Procedia Information Technology & Computer Science 1 (2012) 566-571.

14. Gorp P., Model-Driven Development of Model Transformations, in: Graph Transformations, Volume 5214 of the series Lecture Notes in Computer Science pp. 517-519.