International Journal of Software Engineering and Its Applications
Vol.8, No.2 (2014), pp.375-384
http://dx.doi.org/10.14257/ijseia.2014.8.2.36

Design for U-health Care Hybrid Control Systems

Haeng-Kon Kim and Hyun Yeo*

School of Information Technology, Catholic University of Daegu, Korea
[*]Department of Information and Communication Engineering,
Sunchon National University, Korea
hangkon@cu.ac.kr, yhyun@sunchon.ac.kr

Abstract

In medical healthcare system, when we are supposed to take right decision in a right time, reinforcement learning agent, by combining different model and actions, can help a physician to find out the best diagnosis and treatment options in different states of the patient.

In this paper, the design of a model base for u-healthcare hybrid control systems incorporating sequential and continuous elements is presented. The model base uses various UML concepts to provide an object-oriented description of u-healthcare hybrid systems : Class Diagrams describe the hierarchical relationship between object classes within the same hierarchy; Uses-case diagrams, State Charts and Activity diagrams define procedures for the simulation of UCHCS(U-health Care Hybrid Control Systems) and equations; Collaboration diagrams model the flow of information between objects within the same hierarchy. The model base serves as a template for implementing the hybrid simulator which incorporates equation and UCHCS simulators. The equation and UCHCS simulators are implemented as processes which are executed concurrently using the technique of multi-threading.

Keywords*: U-healthcare Hybrid Control Systems, UML Modeling, Sequential Control, Interactions, Simulation, Component Based Development, Software Reuse*

1. Introduction

Ubiquitous Healthcare is the delivery of health related services and information via ubiquitous computing technologies. It may be as simple as two health professionals discussing a case over the telephone, or as sophisticated as using satellite technology to broadcast a consultation between providers at facilities in two countries, using videoconferencing equipment or robotic technology. Ubiquitous Healthcare is an expansion of the functionality Telemedicine and it encompasses Preventive, Promotive and curative aspects. In our lab, there are various topic about ubiquitous healthcare.

In order to access the timely information and to employ correct diagnosis at anytime and anywhere, use of ubiquitous technologies is becoming ideal test-beds for u-Healthcare environments. However, using ubiquitous device, it would be one of the most crucial requisites to accumulate accurate signals timely and appropriate processing of those signals during such critical circumstances. Furthermore, lack of proper decision support system may delay the treatment, and it may cost a life of the patient. The effort to rectify any of these issues will minimize the time lag between observation and treatment during the emergency circumstances, and helps to reduce the diagnosis time, that can be better utilize for caring the patient.

ISSN: 1738-9984 IJSEIA

Control engineering provides well established methodologies for modeling systems which are either completely continuous or discrete, but no systematic approach exists for the modeling of hybrid systems incorporating both types of elements.
The absence of such a unified modeling technique makes the design of hybrid simulators considerably more difficult than continuous or discrete simulators. This difficulty derives from three fundamental factors:

- **Model complexity**: mixed-mode modeling requires a variety of paradigms and constructs. These include equations, transfer functions, UCHCSs, block diagrams and schematic diagrams.
- **Interactions**: the continuous and sequential components of the hybrid system do not function as separate, autonomous sub-systems, but they interact with each other.
- **Concurrent simulation**: a hybrid simulator must necessarily, incorporate continuous and discrete simulators which must run concurrently to inter-change information.

The problem of model complexity is best addressed by object-oriented modeling which allows diverse model elements to be integrated into a coherent and unified structure. However, object-orientation is a very generic concept. Its application to a specific domain requires a design tool. UML is an efficient tool for the design of object-oriented model bases. Interactions can also be modeled as objects which specify the interacting elements and variables. Concurrent simulation can be implemented either on two inter-connected computers or as multi-task processes running on the same computer. The option used in this project is that of multi-tasking.

2. U- Healthcare Hybrid Control Systems

Ubiquitous healthcare (u-Healthcare) is an emerging paradigm in the healthcare environment. One of the most promising applications for u-Healthcare is the ubiquitous home health monitoring system. This paper addresses two significant challenges in the successful application of the ubiquitous home health monitoring system: the uniform integration of measured biosignal data and easy access to monitored biosignal data. Figure 1 show the u-Healthcare related control systems big picture [1].

Figure1. U-healthcare related Smart Control

Industrial U-healthcare Automatic Control systems are of two main generic types – continuous systems, characterized by continuous-time variables, and sequential systems, characterized by Boolean variables. Continuous systems are, primarily, modeled as

differential equations, but feedback control theory has developed more convenient system-oriented modeling tools such as block diagrams and transfer functions. Sequential systems, on the other hand, are modeled as Sequential Function Charts referred to by the French acronym UCHCS (U-health Care Hybrid Control Systems). UCHCS is the most complete and convenient formalism for modeling sequential control systems [2].

Systems which incorporate both continuous and sequential sub-systems are referred to as u-healthcare hybrid control systems. The modeling and analysis of such systems requires a combination of techniques and paradigms used for both continuous and sequential systems. However, the continuous and sequential elements of a hybrid system cannot be modeled as separate, independent entities since they interact with each other. The problem of interactions is the most fundamental in the modeling and analysis of hybrid systems.

3. UML Modeling of Hybrid Control Systems

UML is well adapted to u-Healthcare Automatic Control Systems if good use is made of meta-models [3, 4). In this paper, a bottom-up design approach is used, starting from specific instances of objects to the more generic classes. UCHCS elements are modeled as classes. The Actions associated with Steps are modeled as Processes (threads). Transition Conditions are also modeled as processes. A Process, referred to in Java as Thread, can be executed independently within a program and this enables several computational activities to run in parallel. Consequently, a « multi-thread » application controls the simultaneous execution of several « threads ». The multi-thread architecture is of fundamental importance in this project since it is often necessary to control or modify one of the constituent components of a control system, without affecting the others. To achieve this:

- a process is linked to a UCHCS as a mechanism of initiating and monitoring the simulation of the UCHCS
- an Action is linked to several processes which are executed simultaneously or sequentially.
- a super-process (also linked to an Action) co-ordinates the execution of all the sub-processes linked to the same Action
- a Transition Condition is linked to a process which computes the logic state of the transition condition and sends a message to the corresponding transition (clearing of a transition).
- every transition is linked to a process which receives messages about the logic state of the transition condition, determines the Steps to be activated, activates the appropriate Steps and deactivates the Steps which were previously active.

Note that some processes might be complementary or incompatible in the sense that the activation of a process might activate other processes or suspend the execution of some active processes.

A profile of some of the classes created for the implementation of the hybrid simulator is presented in Figure 3. Only a single class (FUNCTION) is indicated for continuous systems. The other four classes are for sequential systems.

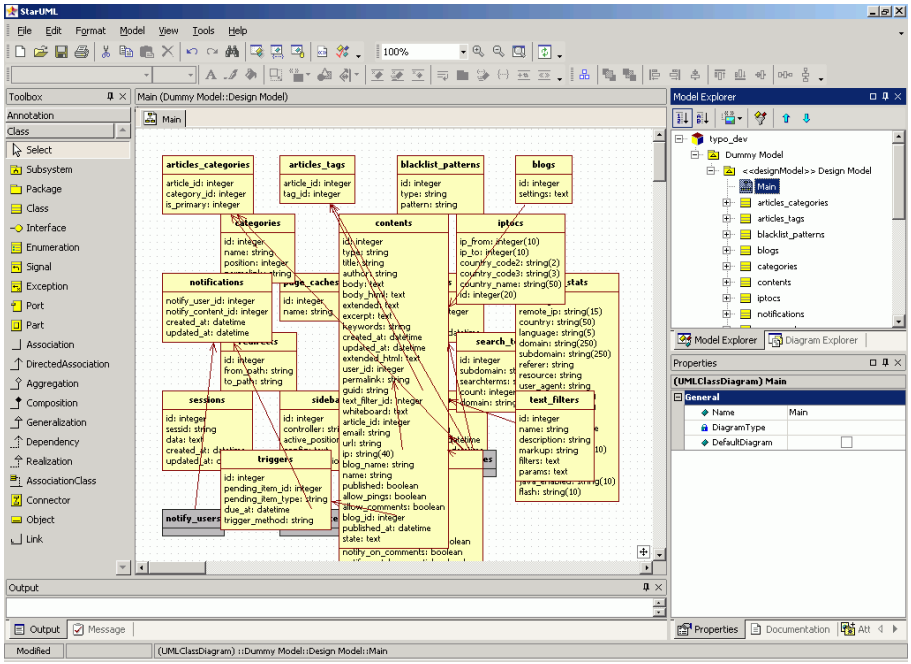


Figure 2. A Profile of Classes for U-Healthcare Control

We model the U-healthcare Hybrid system with UML as Figure 4 through Figure 12. It consists of Use Case Diagram for Specifying UCHCS Parameters, Class Diagram for Sequential Systems and so on.

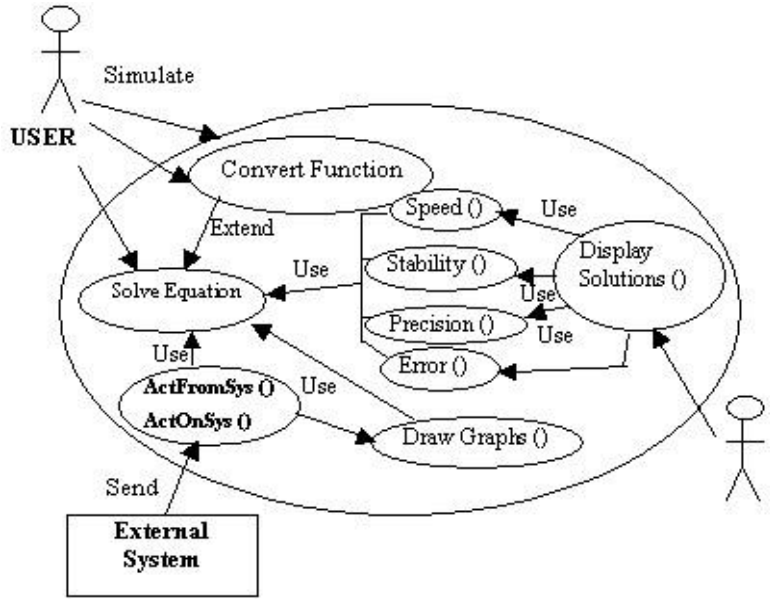


Figure 3. Use Case Diagram for UCHCS

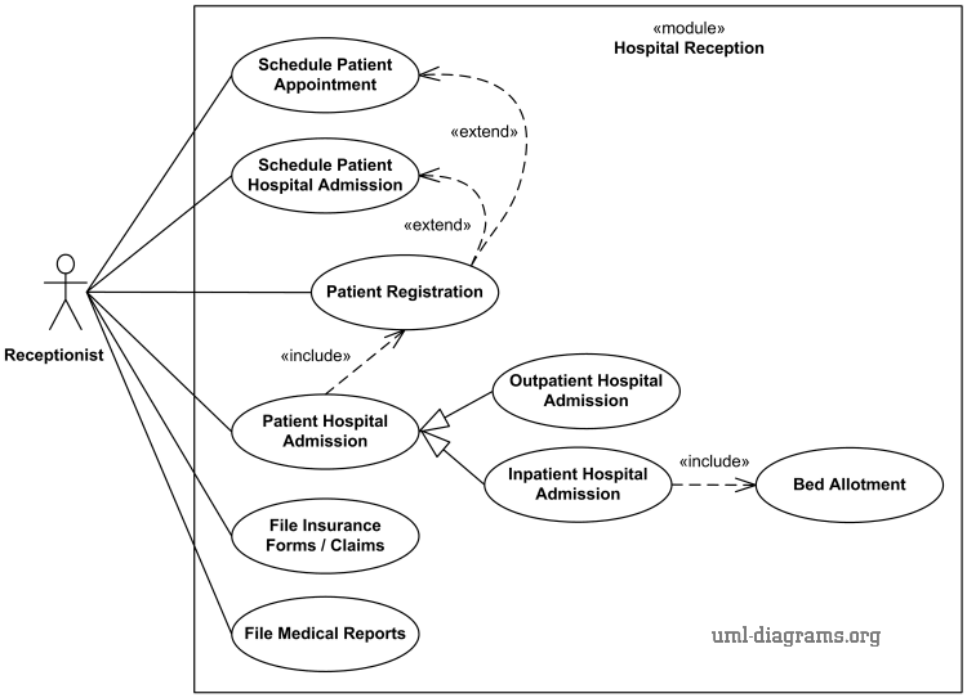


Figure 4. Collaboration Diagram for the Simulation of UCHCSs and Interactions

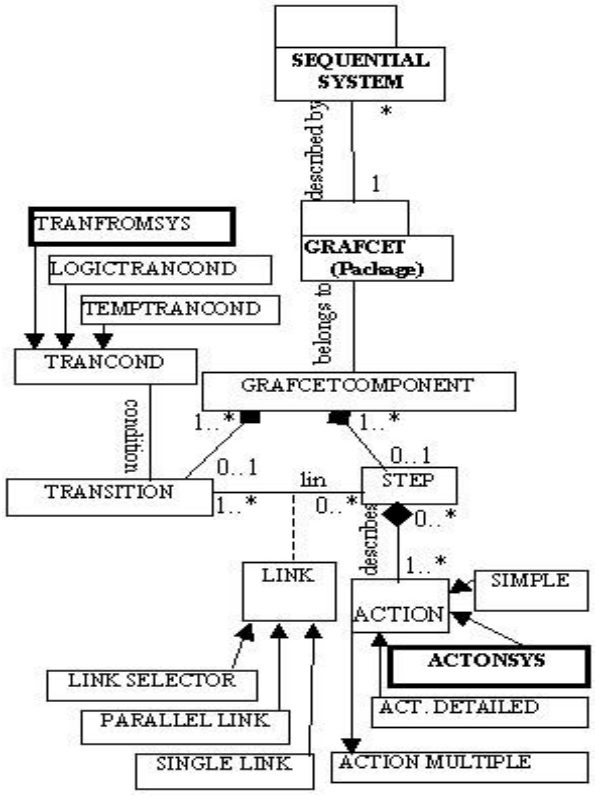


Figure 5. UCHCS Class Diagram for Sequential Systems

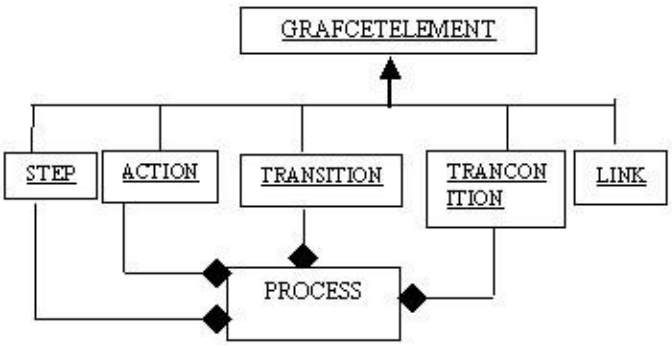


Figure 6. Class Hierarchy Showing Relationship between UCHCS ELEMENT and PROCESS

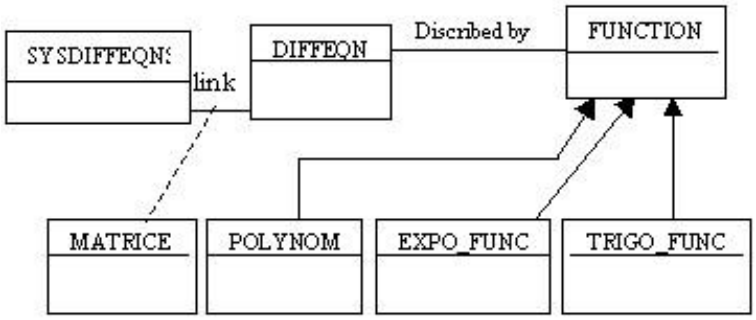


Figure 7. Class Diagram for Mathematical Functions

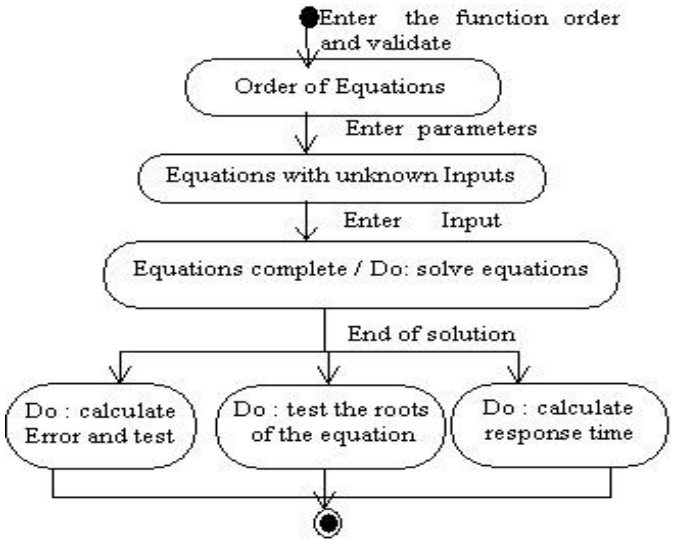


Figure 8. State Chart for Simulation of Continuous Systems

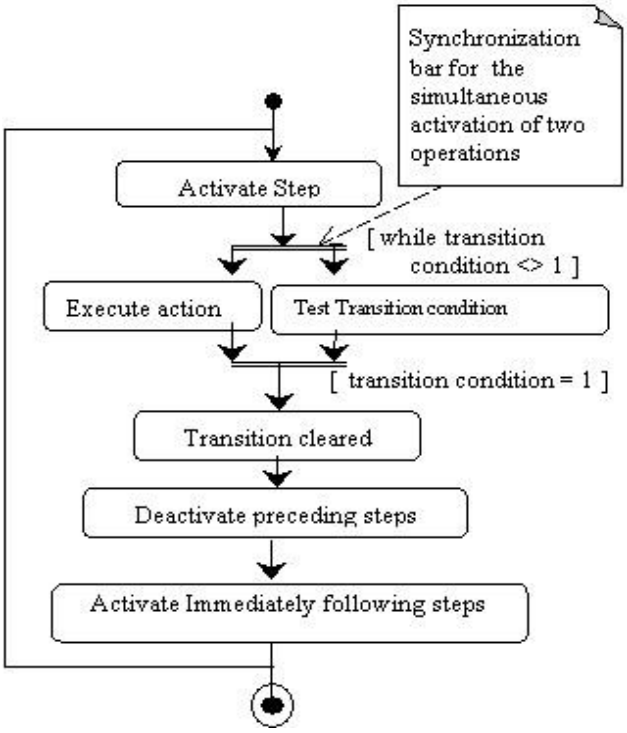


Figure 9. Activity Diagram for the Simulation of a UCHCS

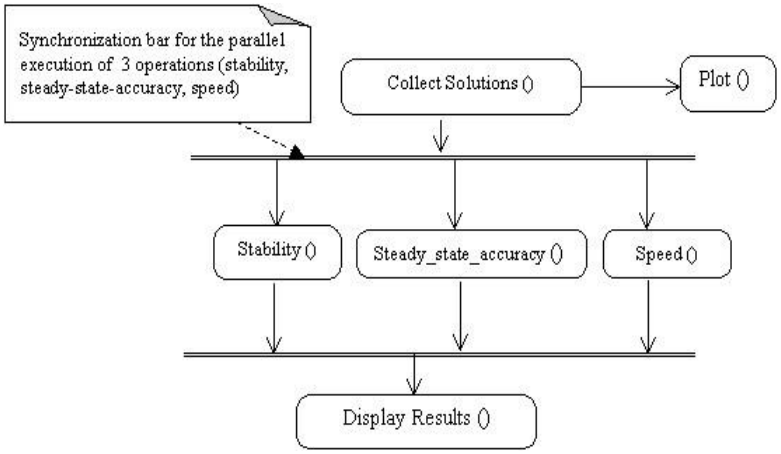
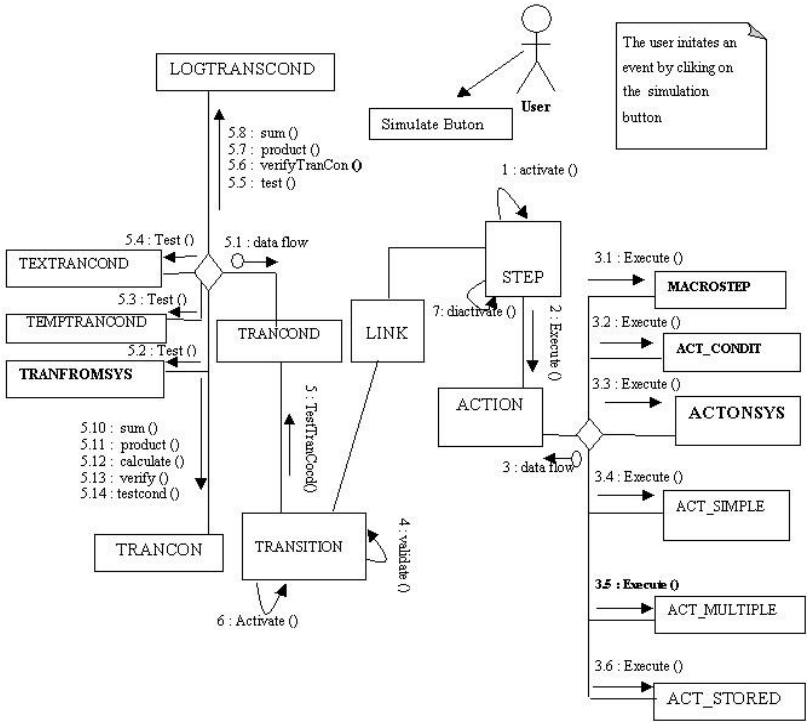


Figure10. Activity Diagram for Continuous Systems



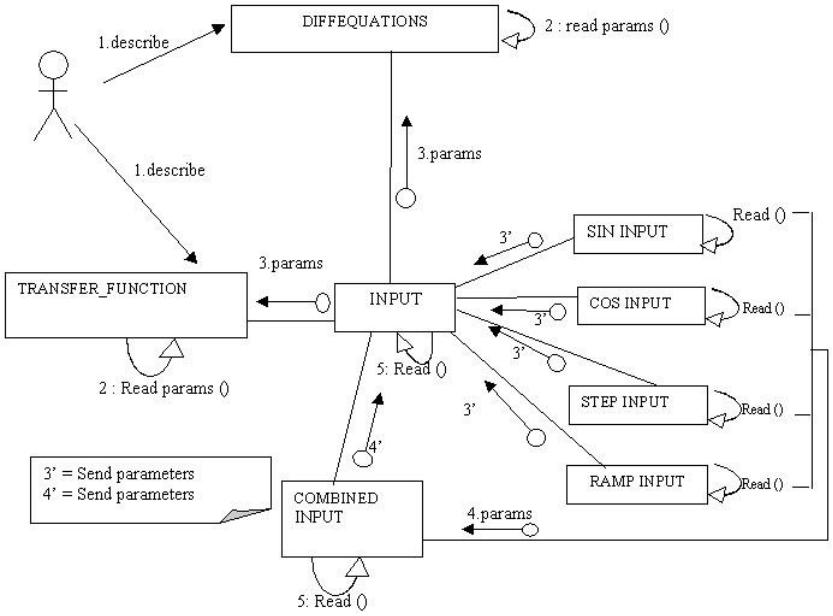


Figure 12. Collaboration Diagram for Continuous Systems

4. Implementation of the Hybrid Simulator

We have completed the development of UCHCS based on the framework repository. We developed the modules defined in the earlier sections: It provides two kinds of transmission modes: immediate transmission and integrated transmission. The former mode operates if a measurement exceeds a predetermined threshold or in the case of an emergency. In the latter mode, the gateway retains the measurements instead of forwarding them. When the reporting time comes, the gateway extracts all the stored measurements, integrates them into one message, and transmits the integrated message to the MS. Through this mechanism, the transmission overhead can be reduced. On the basis of the proposed gateway, we construct a u-healthcare system comprising an activity monitor, a medication dispenser. The running environment of the toolkit is shown in Figure 13 and the explanation of important modules is as follows.

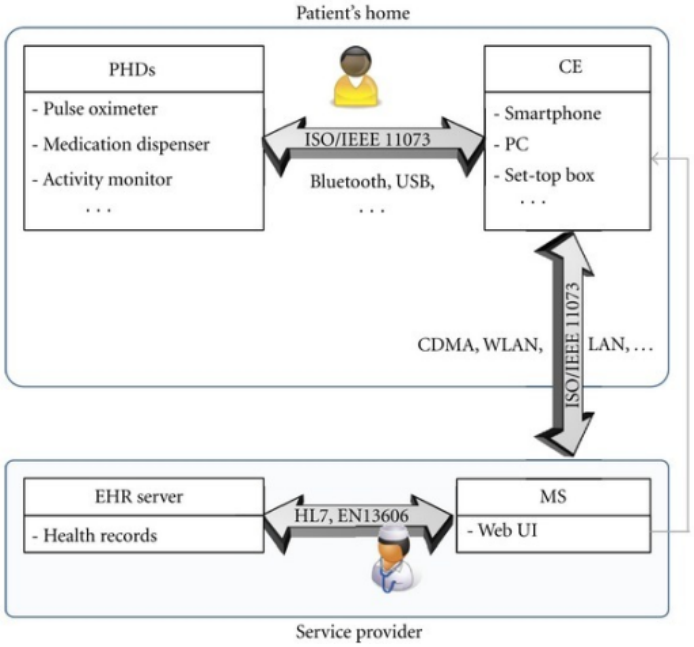


Figure 13. Architecture of *UCHCS*

During the implementation, close reference was made to the UML model base to ensure that the objects and object classes created reflected the required structure. UCHCSs are simulated by indicating the sequence of activation of its steps. An active step is highlighted by displaying it in colour and placing a dot inside the step symbol. Continuous systems are simulated by displaying graphs of output variables and tables of performance indices such as response time, steady-state accuracy and stability margin. The programming of the hybrid simulator was based on the use of THREAD packages to create processes in JAVA. This technique enables multiple processes to be executed concurrently within the same environment. A process was implemented for the UCHCS simulator and another one for the equation simulator. Figure 15 show the UCHCS frameworks prototype.

5. Conclusions and Further Work

In order to access the timely information and to employ correct diagnosis at anytime and anywhere, use of ubiquitous technologies is becoming ideal test-beds for u-Healthcare environments. However, using ubiquitous device, it would be one of the most crucial

requisites to accumulate accurate signals timely and appropriate processing of those signals during such critical circumstances. Furthermore, lack of proper decision support system may delay the treatment, and it may cost a life of the patient.

A model base of u-healthcare hybrid control systems has been designed using the UML meta-model. The model base uses UML constructs such as Class Diagrams, State Charts, Use Case Diagrams, Activity Diagrams and Collaboration Diagrams to model UCHCSs and continuous systems which are the constituent sub-systems of hybrid systems. The model base is used as a template to implement the hybrid simulator in Java. The implementation consists of packages and processes which are executed concurrently using the technique of multi-threading.

Acknowledgements

"This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the CITRC(Convergence Information Technology Research Center) support program(NIPA-2013-H0401-13-2008) supervised by the NIPA(National IT Industry Promotion Agency)".

"This research was also supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning(Grant number: K 2012057499)".

References

[1] N. Kim, Y. Jeong, S. Ryu and D. Shin, "Mobile Healthcare System based on Collaboration between JADE and OSGi for Scalability", (IEEE 2007).
[2] FIPA specifications: http://www.fipa.org/.
[3] http://people.ubilife.net/teemu/?page_id=10.
[4] F. Lo Piccolo, G. Bianchi and S. Salsano, "A Measurement Study of the Mobile Agent JADE Platform", (IEEE 2006).
[5] IBM PCC: http://www.zurich.ibm.com/pcc/.
[6] MS Research Integrated Systems : http://research.microsoft.com/is/.
[7] Intel Personal Health: http://www.intel.com/healthcare/personalhealth/.
[8] Oracle and Toumaz: http://www.toumaz.com/healthcare.
[9] IEEE DS Online-Event Based Middleware Community http://dsonline.computer.org/.
[10] H.-K. Kim, H.-J. Shin, J.-H. Kil and S.-W. Kim, "The Study of the Testing Component for CAI System", Proceedings of the 14th KIPS Fall Conference, vol. 7, no. 2, **(2000)** October, pp. 1337-1340.
[11] KCSC, Component Specification, http://www.component.or.kr, **(2001)**.
[12] J., G. and J., "Software Reuse", Addison-Wesley, **(1999)**.
[13] M. Fowler and K. Scott, "UML Distilled 2", Addison-Wesley, **(2000)**.
[14] B., R. and Jacobson, "The Unified Modeling Language User Guide", Addison-Wesley, **(1999)**.
[15] L. Lemay and R. Cadenhead, "Teach Yourself Java 1.2", SAMS, **(1999)**.
[16] M. Aoyama, "New Age of Software Development: New Component-Based Software Engineering Changes the Way of Software Development", 1998 International Workshop on Component-Based Software Engineering, ICSE, **(1998)**, pp. 124-128.
[17] C. Szyperski, "Component Software: Beyond Object-Oriented Programming", Addison-Wesley, **(1998)** January.

