Remote Patient Monitoring System Architecture for Diabetes Management

Barroon Isma'eel Ahmad
Department of Mathematics
Ahmadu Bello University
Zaria, Kaduna State
Nigeria.
barroonia@yahoo.co.uk

Media Anugerah Ayu
USBI Sampoerna University
Jakarta 12780
Indonesia
media@ieee.org

Ibrahim Abdullahi, SMIEEE
Mathematics/Computer Science
Ibrahim Badamasi
Babangida University Lapai.
Nigeria
ibrojay01@ibbu.edu.ng

Yussuf Yakubu
Computer Sci. Department
FCE Zaria,
Kaduna Nigeria
yussyak@gmail.com

***Abstract*—Health Information Technology (HIT) is becoming popular these days because of its immense benefits such as improving quality, efficiency and healthcare delivery. HIT can be used to provide medical services to patients from a distance typically via Internet or mobile phone services like SMS. Similarly, remote patient monitoring is a form of HIT that brings together the fields of information technology and medical sciences. Remote patient monitoring allows patients suffering from different diseases particularly chronic to be monitored, in order to have constant care. With remote patient application cost involved in managing a disease can be reduced and at the same time overcome the limitation of lack of specialists, resources, etc. Remote patient monitoring will be more interactive if it utilized Electronic Health Record (EHR) and Clinical Decision Support System (CDSS), because very often a response or decision is required when patients send their data to the healthcare centre in case of emergency. Also time spent on patients will be reduced, since the system will help doctors in making decision about their patients. However, nearly all the available remote monitoring applications are based on architectures that do not include CDSS feature, or are based on architectures that separate between Electronic Health Record (EHR) and CDSS module which are normally associated with interoperability problems. In this work we propose a complete remote patient monitoring architecture which uses EHR to make decision about patients suffering from diabetes.**

***Index Terms*—Clinical decision support system, remote patient monitoring, knowledge base, diabetes mellitus, electronic health record, layered architecture**

1. Introduction

Diabetes Mellitus is a chronic disease that affects about 347 million people (about 5 percent of the world's population) out of which more than three-quarters live in developing countries. People suffering from diabetes need to be monitored, since their blood sugar ought to be regulated. However, there are several remote diabetes patients monitoring systems but mostly they cannot make decision about the patients; this is very essential if we look at the challenges faced in developing countries like lack of qualified medical personnel and health care facilities among others [1]. On the other hand, to develop any system, application, or software the following system development stages need to be followed, these include identifying the requirements/specifications, design, development, testing and finally implementation [2, 3]. In the course of the design, the architecture of the system need to be defined first then the system will be designed based on that architecture [4, 5]. The architecture is the abstract representation of the subsystems, and the flow of information within subsystems that make up the whole system [5, 6]. In this work, we are proposing an architecture of a system that would help doctors to make decision on their patients; these types of systems are commonly referred to as Decision Support Systems.

Decision Support System (DSS) is an information system that supports decision making activities in an organization, quite often DSSs are interactive [7]. A DSS that focus on clinical applications is called Clinical Decision Support System (CDSS). CDSS as described by Jao and Hier [8], is a system that uses case-based reasoning to help clinicians access disease status, make diagnosis, select appropriate therapy or make other clinical decisions. The use of words like support, help, etc. in defining DSS is very important, especially to clarify the fact that DSS are not replacing the experts or decision makers [9]. Generally, DSS application consists of five basic components but the content and structure may vary, these are: database, model base, knowledge base, user interface and users [10]. The database stores the data that is used for the decision making, it also allows the user to access and manipulate the data. Model base allows the analysis of the problem using models like financial, statistical, etc. Knowledge base is like a repository where knowledge about the problem is kept; usually it represents the expert in a particular field. User interface deals with communication between the user and the DSS. Users are the individuals that use the DSS; in other word are the decision makers [11].

Several types of DSS exist, they include not limited to: text-oriented, hypertext-oriented, database-oriented, spreadsheet-oriented, rule-oriented and so on. The difference is normally in terms of knowledge system representation and the way the system process problems [12, 13]. Similarly, DSSs are classified as either standalone or web based. Although the availability of Internet has positively affects the use of many computer applications, because users can access them from anywhere and share information concurrently; but DSS applications are still largely standalone. Making DSS available on Internet will of course enhance the organizational decision making process, so we can have a DSS that can be accessed on WWW [10]. This type of DSS is called web-enabled DSS. The requirement for successful web-enabled DSS are: database management system like MySQL, Microsoft access for storage and management of data; programming language like HTML, Java, VB.NET for GUI design, data processing etc.; scripting language like PHP, ASP.NET for web-enabling [10].

Nevertheless, the DSS regardless of being it standalone or web enabled are built based on a particular architecture. The architecture in a broader definition by IEEE standard is The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution. [14]. So we can view our proposed architecture as an outline of the system with its components as well as the properties and relationship between them. The proposed architecture is built with the idea of software and DSS architecture along with expert system concept [15].

Currently, the main drawback of most of the patients monitoring systems is the lack of decision making capability, or they are autonomous systems separated from decision support systems [16-18]. In this work we are propose an architecture of a patient monitoring system, coupled with decision support feature.

2. Literature Review

A lot of work has been done on the area of decision support system architecture, ranges from an architecture in a specific area to a generalized architecture, but very few focus on issues related to integration into HIT applications or developing such architectures.

Research in the area of DSS mostly focuses on the architecture of either DSS or CDSS as an autonomous software or a software module incorporated into Electronic Health Record (EHR). For instance, Lv and Li [19] proposed an architecture of DSS that integrates data warehouse, knowledge warehouse and model warehouse as a foundation for the DSS system; this can be seen as a work on general architecture of DSS. CONNECT (Care Online: Novel Networks to Enhance Communication and Treatment) framework is a service architecture framework that was proposed and implemented by Mirkovic, et al. [20]. Session management, content adaptation and mobility issues (like session transfer, hand-off between networks) are the main concern in developing the framework.

CDSS Architecture in Korea is an article that presents an architecture of CDSS in view of EHR. How to integrate CDS service with HIS was suggested, since the CDS architecture described is an autonomous service [21]. They consider knowledge engine as a very important part in implementing knowledge service, hence they also suggested interoperability between CDSS and knowledge engine architecture. Hypertension guideline application was developed to validate the knowledge engine and CDSS architecture. Another CDSS architecture was proposed by El-Sappagh and El-Masri [22]; which uses data mining techniques to build cooperative knowledge bases from domain experts knowledge bases, clinical databases, and most recent academic researchers. The data mining engine is connected to the EHR and clinical databases to continuously mine the very recent knowledge and adds it to local knowledge base, specialized knowledge bases from other institutions can also be consulted for relevant knowledge. In this architecture, distributed knowledge bases for specific diseases will be connected to distributed EHR platforms from different hospitals [23].

Khalid, et al. [24] present a model of a cost effective health care system for patients residing in remote areas of Pakistan. The complete architecture of the system consists of wearable medical sensor module, data gathering module, PDA, remote server with CDSS and Electronic Medical Record (EMR) capability, and web enabled remote terminal for accessing services provided by web server. The remote server after processing the data then call CDSS for analysis of the data and finally the EMR will record the data against the patient profile. After analyzing the data by the CDSS a feedback is sent to the doctor for approval, and then sent to the PDA after approval. The CDSS software analyses the patient physiological data like ECG, blood pressure, temperature, etc. for possible sign of abnormality. The software can project the health status based on the received data and also can make decision based on the health situation.

A combination of model-driven and knowledge driven decision support systems were used. The model-driven makes decision based on the statistical model of the patient data, while the knowledge driven use facts, rules, procedures, etc. to make decision. For diagnosis a cooperative system is employed in which the decision and action is first send to the consultant for confirmation before sending to the PDA. The EMR store the patients data and serve as a source for data to CDSS. When new data is received the CDSS will get previous from the EMR and use it as a basis for analysing the current data, then store it with the result of the analysis in the EMR. On the other hand, Gomes, et al. [25] present a DSS that will help surgeons and hospital managers to schedule patients as well as allocate resources. Web service is used for integrating DSS with HIS; a third party integration agent called AIDA was used as a communication layer of the HIS. Enterprise oriented architecture was used to divide the software into four layers: data access; business logic; web service; and presentation layers. To synchronize DSS and HIS, an update service calls a web service in AIDA for a request to synchronize data warehouse with recent data in HIS then the update service request the shared database to update the DSS database.

However, Boreisha and Myronovych [26] discuss about creating web-based DSS using Web services. The main components of their DSS are database, user interface, and DSS software system. Three layer design, Rich Internet Applications (RIA) and web services are the core elements that made up the web based DSS. The layers are data, business logic, and view. The RIA are characterized by their performance which is derived from AJAX and XML and strong GUI, this makes web applications more responsive. The web services are web based applications created as a module with interfaces for linking to other applications normally via Internet. The main advantage of using web service is ability to integrate into applications regardless of their platform or operating system. In their discussion the web service will provide interface between AJAX enabled user interface and the DSS. So the web service sits between the DSS software system and User interface.

Chakravarti and Bhattacharyya [27] present an architecture as a solution for CDSS using hand held devices. The architecture of the system has five modules. Analytical engine: responsible for analyzing patients data and send alerts, warnings and recommendations. Transaction engine: respond to clinicians request by finding the relevant data and display the result. Supplemental engine: retrieves clinical data clean it and prepare it for data warehousing. Security module: ensures the confidentiality and privacy of the data. Interfaces: link the system with HIS, data warehouse and wireless network.

Similarly, the idea of integration can be seen from different perspective, some researchers view integration not as integrating DSS into other applications but integrating some of the DSS components together. For instance a work by Liu, et al. [28] who presented five different integration types. They are:

1. Data and information integration: this involves obtaining reliable information without considering how data and information is operated on and transformed by DSS components, so as to have a DSS that users can depend on. This deals with data and information representation, conversion, storage, etc.
2. Model integration: this has to do with integrating sub-models to come up with a large cohesive model. Having different view of the real problem help to have a more balanced decision making.
3. Process integration: DSS is considered well integrated in terms of process if two or more of its components have uniform assumptions about a process.
4. Service integration: this involves a component providing service to other components that they need, it also involves knowing which component will use its functionality. It also deals with how to control and share services.
5. Presentation integration: this involves using appearance and behavior integration to make it easy for users to interact with a component using the experience from another component.

The five integration types can be further improved by using knowledge based systems, data mining, intelligent agents and the web technology. Likewise, Ongenae, et al. [29] proposed a framework that integrates a rule-based component into existing Service Oriented Architecture (SOA) platform. The integration was done and evaluated in two ways, database-driven and service-oriented approach. In the database-driven approach the rule-based system (stand-alone application) is added in between the database and the services as an additional layer, while in service-oriented approach the rule-based system (component) does not interact with the database directly but rather works with the data provided by the services, so its fully integrated into SOA.

Another integration related work is presented by Verlaenen, et al. [30], that integrates different clinical decision support models. Clinical knowledge models are used to model clinical knowledge in order to create guideline in a computer-interpretable format. Clinical workflow, clinical pathways, validation rules, and Proforma are the four different models used for implementing the prototype.

From the works above, its clear that the developed architectures were mainly standalone as either an application on its own or a software component in EHR. Though some researchers worked on integration, but they usually focus on integrating DSS components rather than integrating DSS into EHR. Our proposed architecture is a complete EHR architecture with decision making capability.

3. Proposed EHR/CDSS Architecture

EHR is rapidly changing healthcare industry by improving the quality of care and cutting down healthcare cost. Integrating CDSS into EHR has the potential to enhance patient monitoring by providing services such as alerts and reminders [31-33]. This paper present an architecture of EHR/CDSS for remote patient monitoring system. The proposed architecture will address the challenges of the existing architectures like need for interface, inter process communication, etc. [34, 35]. The architecture can also benefit from the common advantages of using layers in architecture design like ability to make changes in one layer without affecting other layers; easier to maintain since the functionalities, data, domain, application and presentation are separated. For example change in the model can only affect domain layer, like wise a modification in the user interface can only affect presentation layer.

3.1. Logical Architecture

Enterprises 3-layer and web based 4-layer architectures are the foundation of our logical architecture [36]. The difference is that the domain and application layers from web based architecture are merged into application/business layer in our architecture, which makes it appear like enterprises architecture. Then additional layer is introduced, even though it is called domain but it is not the actual domain from the web based architecture. Our domain architecture is a CDSS module comprises of knowledge and model bases. Knowledge-based CDSS is the type of CDSS considered in the architecture design. The architecture is shown in Figure 1 and 2.

3.1.1. Data Layer. This layer communicates with domain and application/business layer. In a three layer architecture, normally this layer communicates with either domain or application layer only, but in our design the layer communicates with both two not at the same time but depend on the situation. For example, if the system is accessed by the patient, doctor or administrative staff just to retrieve information related to perhaps personal information of the patient, the application layer will get the information from the data layer without passing through the domain layer because there is no need for making any decision. But on the other hand if the system is accessed for the purpose of treatment or anything that requires decision making, in this case the application layer will have to first communicate with the domain layer; and then the domain layer will get relevant information from the data layer and make the appropriate decision. Proper security has to be put in place to ensure that this layer performs as expected.

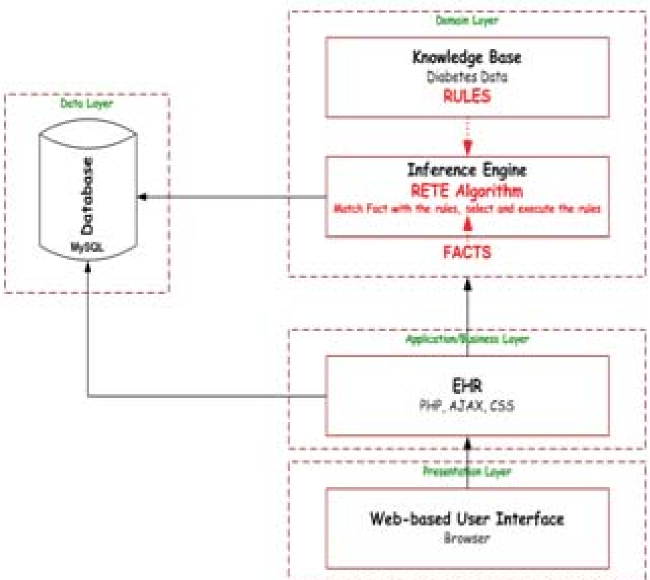

Figure 1. Logical Architecture

3.1.2. Domain Layer. In this layer the core decision functionality takes place, this is where the logic and the like will reside. The layer has two important modules, knowledge base and model base (inference engine). The knowledge base is where the facts and rules about diabetes treatment will be stored. These rules are what will be used when there is need for decision about the patient. Generally, the knowledge about diabetes is stored here, and its assumed that a new knowledge rule can be added or updated. The knowledge in this case is stored as rules, so our rule base is the knowledge base. The rules are of the form: IF condition (s) THEN action (s). The input data from the user or system generated will serve as the condition while the action is the output or an input to another rule. Typical example is when the patient send like blood glucose concentration value of 200 mg/dl, the decision will be to consult nearest clinic because it is beyond the normal value: IF blood_glucose $>$150 AND $<$ 70 THEN alert(Visit nearest hospital or clinic). The model base or inference engine is what processes the input against the rules stored in knowledge or rule base. This layer communicates with data as well as application layers, it does not have direct link with the user or presentation layer but rather the user will communicate with application layer first and the application will decide if it needs to establish connection with the domain layer.

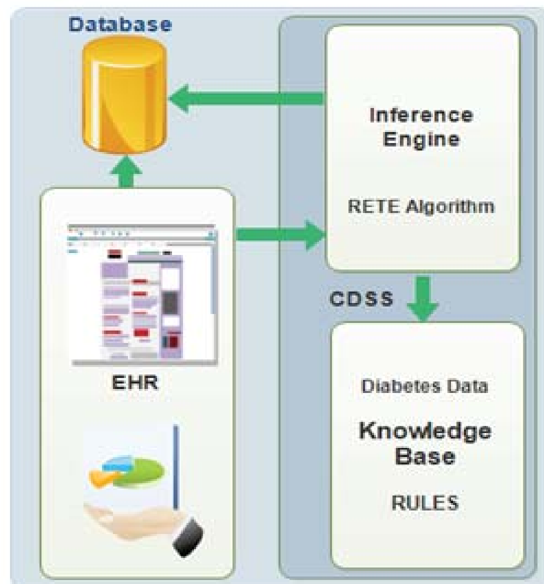

Figure 2. RPMS Architecture

3.1.3. Presentation layer. This layer contains the user interface for the system. The interface will be designed by being aware of the fact that the users can access the system using PCs or mobile devices. In the patient side, the interface will be more complicated since they can attach some medical devices to it and they may need some graphs for easy interpretation of their results and progress. Also the patient interface will be made simple since the users may not necessarily be educated or computer literate but should be made such that at least the mobile interface can be easily used by them. Another level of security will be implemented in this layer to take care of authentication and authorization.

3.1.4. Application layer. This is the layer that takes care of managing the information for the whole system. The information can be patient information like their basic information, diagnosis information, treatment information, etc. or doctors information like their personal information, the patients assigned to a particular doctor, etc. Information about the users of the system like login credentials is also managed by this layer. However, the layer is responsible for generating reports, statistics, graphs, etc. about patients. Sending an alert, reminder, etc. are all function of this layer.

3.2. Physical Layer/Tier

The physical architecture is developed using 3-tier architecture concept. Application and domain layers are combined in one physical system (tier), while the data and presentation layers are each on separate tier. The choice of combining application and domain layers in one tier is, to increase the performance of the system by eliminating inter system communication between the two layers. Figure 3 depicts the physical architecture. The physical and logical architectures are quite similar in terms of the functionalities, except that in physical architecture the tiers are assigned into physical system (computer).

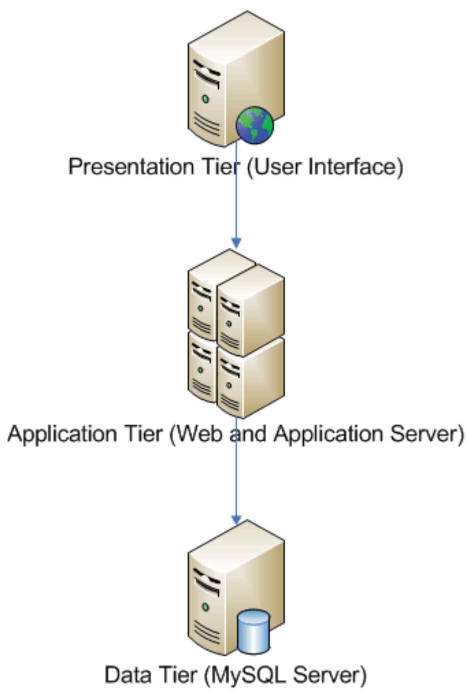

Figure 3. Physical Tier

3.2.1. Presentation tier. This includes the user interfaces that display data to the user and accept input from the user. The data that will be displayed can be in the form of graph, chart or in basic html, PDF formats.

3.2.2. Application tier. This is the heart of the system, because the fundamental functionality resides in this tier. The application layer manages the information about the users particularly the patients, while the domain takes care of the decision making.

3.2.3. Data tier. This handles the communication between the application tier and the database. The database is the only application that is run on this system; this will make the data more secured, since the users lack direct access to the database.

4. Conclusion

An integrated EHR/DSS architecture for remote patient monitoring is proposed in this paper. The architecture presented is promising with detailed descriptions on how it shall overcome problems associated with integrating CDSS into EHR. As presented in the sections, succinct elaborations were captured as it relates to all the component of software architecture, DSS as well as expert system which will be used to implement the architecture. This architecture was developed with chronic disease patients particularly diabetic patients in mind. Therefore, when the architecture is implemented it will allow the patients to be monitored remotely from sparse or close geographical dimension. This will consequently increase access to health-care; reduce hospitalization, decrease health-care cost. In the future, a remote patient monitoring system would be developed from this architecture, and implemented in a real clinical setting.

References

[1] World Health Organization. (2012, September 14). Diabetes. Available: http://www.who.int/mediacentre/factsheets/fs312/en/

[2] J. Jlinoja, "Requirements implementation in embedded software development," University of Oulu, Department of Information Processing Science, Masters Thesis, p. 85, 2004.

[3] Z. D. Ulbert, Software development processes and software quality assurance: University of Pannonia, 2014.

[4] L. Jocic, R. Herm, M. Jacobs, A. Amram, E. Sundberg, I. Min, et al., "Decision Support Framework: Architecture Development," in Aerospace Conference, 2012 IEEE, 2012, pp. 1-12.

[5] L. Maciaszek, Requirements Analysis and System Design. Canada: Pearson Education, 2007.

[6] H. Jaakkola and B. Thalheim, "Architecture-Driven Modelling Methodologies," in EJC, 2010, pp. 97-116.

[7] M. Sethi and A. Sharma, "Information System and System Development Life Cycle," in Software Development Techniques for Constructive Information Systems Design, K. A. Buragga and N. Zaman, Eds., ed: IGI Global, 2013.

[8] C. S. Jao and D. B. Hier, "Clinical decision support systems: An effective pathway to reduce medical errors and improve patient safety," Future, vol. 4, p. 4, 2010.

[9] S. B. Eom. (2001, September 25). Decision Support System. Available: http://cstl-hcb.semo.edu/eom/iebmdssrwweb.pdf

[10] A. A. Pol and R. K. Ahuja, "Developing Web-Enabled Decision Support Systems," Belmont, MA: Dynamic Ideas, 2007.

[11] E. Turban and L. Volonino, Information Technology for Management: Improving Strategic and Operational Performance, 8th ed. New York: Wiley, 2010.

[12] V. L. Sauter, Decision support systems for business intelligence: John Wiley & Sons, 2014.

[13] C. W. Holsapple, "DSS architecture and types," in Handbook on Decision Support Systems 1, ed: Springer, 2008, pp. 163-189.

[14] IEEE, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," ed: IEEE Explore, 2007.

[15] Microsoft, ”Microsoft Application Architecture Guide: Patterns & Practices ” 2009.

[16] S. Mukherjee, K. Dolui, and S. K. Datta, ”Patient health management system using e-health monitoring architecture,” in Advance Computing Conference (IACC), 2014 IEEE International, 2014, pp. 400-405.

[17] P. Vrady, Z. Beny, and B. Beny, ”An open architecture patient monitoring system using standard technologies,” Information Technology in Biomedicine, IEEE Transactions on, vol. 6, pp. 95-98, 2002.

[18] G. Paliwal and A. W. Kiwelekar, ”A Product Line Architecture for Mobile Patient Monitoring System,” in Mobile Health, ed: Springer, 2015, pp. 489-511.

[19] J. Lv and Z. Li, ”Research on architecture of decision support system,” in Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, 2009, pp. 199-202.

[20] J. Mirkovic, H. Bryhni, and C. M. Ruland, ”A framework for the development of ubiquitous patient support systems,” in Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on, 2012, pp. 81-88.

[21] J. Kim, I. Cho, and Y. Kim, ”CDSS (Clinical Decision Support System) Architecture in Korea,” in Convergence and Hybrid Information Technology, 2008. ICHIT’08. International Conference on, 2008, pp. 700-703.

[22] S. H. El-Sappagh and S. El-Masri, ”A Proposal of Clinical Decision Support system Architecture for Distributed Electronic Health Records,” world-comp. org, 2011.

[23] S. H. El-Sappagh and S. El-Masri, ”A distributed clinical decision support system architecture,” Journal of King Saud University-Computer and Information Sciences, vol. 26, pp. 69-78, 2014.

[24] M. Z. Khalid, A. Akbar, A. K. Tanwani, A. Tariq, and M. Farooq, ”Using telemedicine as an enabler for antenatal care in Pakistan,” in 2nd International Conference E-Medisys: E-Medical Systems, Sfax, 2008.

[25] C. Gomes, F. Sperandio, J. Borges, B. Almada-Lobo, and A. Brito, ”A Decision Support System for Surgery Theatre Scheduling Problems,” in ENTERprise Information Systems, ed: Springer, 2011, pp. 213-222.

[26] Y. Boreisha and O. Myronovych, ”Web-based decision support systems as knowledge repositories for knowledge management systems,” UbiCC Journal, vol. 3, 2008.

[27] B. Chakravarti and S. B. Bhattacharyya, ”Mobile Based Clinical Decision Support System,” n.d.

[28] S. Liu, A. H. Duffy, R. I. Whitfield, and I. M. Boyle, ”Integration of decision support systems to improve decision support performance,” Knowledge and Information Systems, vol. 22, pp. 261-286, 2010.

[29] F. Ongenae, T. Dupont, W. Kerckhove, W. Haerick, K. Taveirne, F. De Turck, et al., ”Design of ICU medical decision support applications by integrating service oriented applications with a rule-based system,” in Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on, 2009, pp. 1-6.

[30] K. Verlaenen, W. Joosen, and P. Verbaeten, ”Arriclides: an architecture integrating clinical decision support models,” in System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, 2007, pp. 135-135.

[31] M. Alwan, J. Bravinder, P. Burnstein, E. Eiting, D. Loveland, S. M. Hooper, et al., ”Telehealth and Remote Patient Monitoring for Long-term and Post-acute Care: A Primer and Provider Selection Guide,” LeadingAge, Ed., ed, 2013.

[32] M. Sambasivan, P. Esmaeilzadeh, N. Kumar, and H. Nezakati, ”Intention to adopt clinical decision support systems in a developing country: effect of Physicians perceived professional autonomy, involvement and belief: a cross-sectional study,” BMC medical informatics and decision making, vol. 12, p. 142, 2012.

[33] B. Rothman, J. C. Leonard, and M. M. Vigoda, ”Future of electronic health records: implications for decision support,” Mount Sinai Journal of Medicine: A Journal of Translational and Personalized Medicine, vol. 79, pp. 757-768, 2012.

[34] A. Sen, A. Banerjee, A. P. Sinha, and M. Bansal, ”Clinical decision support: Converging toward an integrated architecture,” Journal of biomedical informatics, vol. 45, pp. 1009-1017, 2012.

[35] D. F. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, et al., ”Grand challenges in clinical decision support,” Journal of biomedical informatics, vol. 41, pp. 387-392, 2008.

[36] L. Kindblad. (2010, August 30). Layered Architecture. Available: http://www.slideshare.net/kindblad/application-architecture.