An Artifact on Prediction of Software Architecture Style in Medical Domain

Umesh Banodha
Department of Computer Applications,
Samrat Ashok Technological Institute,
Vidisha (M.P), India

Kanak Saxena
Department of Computer Applications,
Samrat Ashok Technological Institute,
Vidisha (M.P), India

ABSTRACT
The paper's focus point on understanding of software system with its development on the concept of software architecture styles in the field of medical domain. In order to fasten the software maturity process for patients' recovery and ease the diagnosis even in the lack of the connoisseur hands, an artifact in the form of algorithms which predict the software architecture styles. The proposed algorithms will classify the patients' into different classes that are sound and appropriate for the respective architectural styles. The paper contains three categories of architecture styles namely (i) Communication architecture style (Pipe and filter architecture style) (ii) layered architecture style (iii) Separated Presentation architecture style (Model View Controller). Further the proposed algorithm is tested on the liver data set.

KeyWords
Software Architecture, Quality Attributes, Pop count, medical process reengineering.

1. INTRODUCTION
As all know that over the last 200 years there was major development as well as advancement in medicine and diagnosis maturity levels techniques, but the identification of disease's signs became more and more the doctors' domain. The health care professional focus on signs as well as symptoms and evidence, it is vital too, for the software development process to do so. The purpose is to artifacts the importance of the software architecture in the era of health care due to the following major properties.

1) Simpler the software architecture, resultant in implementation.
2) Improved reusability with the affect of software architecture style.
3) Adaptability and robustness.
4) Above all results in cost savings with the appropriate architectural style selection.[1]

The Software architecture denotes the high level structure of a software system. Software architecture facilitates communication between stakeholders, captures early decisions about the high level design and allows reuse of design components between processes [2]. The problem is how one can understand software system and their developments because (i) there is a big gap in software and traditionally engineered system (ii) differences between the content of architecture views in the respective disciplines and (iii) how these views are used in the development processes in medical era.

Software maturity level is associated with appropriate architecture style. Due to limitation of engineering approach like different types of quality factors we will focus on software architecture style merely. For the medical domain problem, software architecture is being widely used to develop a suitable high level designing for large software system.

Software Architecture or Architecture is the structure of the components of a program or system, their interrelationships and the principles and guidelines governing their design and evolution over time. [3]

The software architecture of a program or computer system is structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. [4, 24]

The importance of software architecture arose at very first step during system development especially in medical domain as they virtually affect every later stages of the maturity process which will direct or indirect impact the mortality rate of the patient at the end. Thus, good software architecture can reduce the risk with building a technical solution and make the system realization and testing more observable as well as achieve higher quality attributes. [5]

Software architecture presents a common abstraction of a medical decision system. The user can use this abstraction as a milestone for perceptive, decision orientation, communication and healthcare with following benefits

- A clear vision and roadmap for the team to follow, regardless of whether that vision is owned by a single person or collectively by the whole team.
- Technical leadership and better coordination.
- A stimulus to talk to people in order to answer questions relating to significant decisions, non-functional-requirements, constraints and other cross-cutting concerns.
- A framework for identifying and mitigating risk.
- Consistency of approach and standards, leading to a well structured codebases.
- A set of firm foundations for the rest of the project.
- A structure with which to communicate the solution at different levels of abstraction to different Patients'. [3, 4, 5,6, 7]

Currently, to sustain the fitness of human is one of the major research areas and, software can play vital role in medical domain. The software brings a revolution in medical area. It imposes the improvement in software maturity, which not only save the life of human but also increases the life span. The software architecture styles plays vital role to identify the severity level of patients' which will fasten the identification process of major signs with verification and validation and hence, significant improvement in patients' health.

Medical domain compile with various nature of active components. A few of them are Patient, Doctors, supporting staff, Experts/Specialist, medical equipments and devices, appropriate software. Studies proved that no components are up to date and up to the mark. Some of them are having lack in on one side so then the other side. This paper mechanism the

association of software architecture styles in the medical era. It expounds several serious issues that have been identified which results in the enhancement of medical processes based on signs and symptoms occurred with the error rate. [6, 8]

Software architecture constitutes a relatively small, intellectually graspable medical model for how a system is structured and how its elements work together that is transferable across system. [9]

The research in medical domain always requires more concentration due to betterment of patients' curability of diseases in comparison of cost and efforts to identify the severity level of patients. [6]

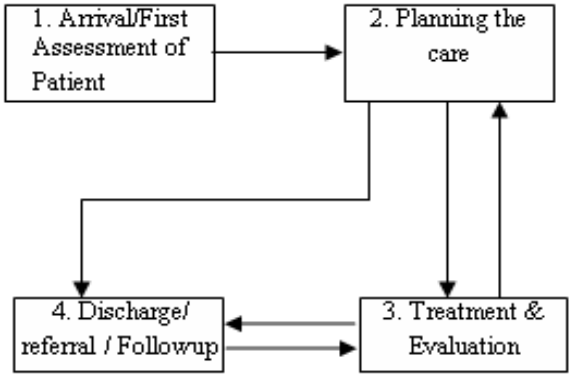


Fig 1 Process Modeling Approach

2. ARCHITECTURAL STYLES

An architectural style is a coarse-grained pattern that provides an abstract structure for a family of systems. It improves partitioning and promotes design reuse by providing solutions to frequently recurring problems and help in the design, analysis, implementation, and evolution of architectures. [22]

The main purpose of a style is to; firstly, documentation of the design and implicitly presents alternatives of the design. Secondly, applying a style means reusing a specification, or a specification template, and lastly, under certain circumstances, not only the specification may be reused, but also the results of a analysis of the style [Monroe et al. 1997].

It is a template for an architectural unit, and may contain all the elements that an architectural unit may contain. In particular, it contains (1) architectural units, or partial specifications, templates or constraints for them (2) behavioral constraints (3) partial specifications of the static and dynamic structure of these architectural units and glue. Finally, (4) a style contains an account of its rationale and usage conditions. All of these aspects are stored in appropriate views of the style. Architectural styles are the composition patterns and constraints on architectural elements and their interrelationships targeted at families of systems with shared characteristics [Abd-Allah 1996, Fielding 2000].

There are many architectural styles currently in use, including batch sequential, blackboard, client-server, and data flow, event-based, layered, model-view-controller, objects, peer-to-peer, pipe-and-filter, and publish-subscribe as well as their many variants .

Styles primarily focus on the following:

- External structure of the allowed architectural elements in the style;
- Types of supported interactions (i.e. software connectors), and their allowed specializations;
- Topological constraints placed on configurations of architectural elements. [22]

The following sub sections demonstrate the use of three architecture styles which are associated with medical era and later tested on liver data set.

2.1 Layered Architecture Style

The layered architecture style is most suitable when domain is huge and complex. In layered architecture style the application is decomposes at different layers i.e. multilayer, which is encapsulated with roles and responsibilities. The decomposition of layers is in hierarchical manner and each layer is strongly coupled to other layers through well defined interfaces. The role of layer means the nature, function, mode and type of interaction with others. The responsibilities mean how functions are mapped with others. The major properties are (i) high level of cohesion, (ii) Loose coupling and (iii) reusability [22].

The layer architecture style of medical domain contains the four layers as in figure No.2 with their roles and responsibilities. The first layer is physical resource layer which reserves the data of patients' i.e. signs of patients' diseases based on the initial symptoms. The second layer is data preprocessor which is responsible for choosing the correct signs and symptoms of patients with the help of data selection process with preprocessing of data. The most significant layer is third layer the medication layer which does diagnosis based on the signs and symptoms of patients' with the help of medical test if required or apply the medication constraints. The last layer is session layer which provides the interaction between medication and patients/external systems. The assessment process is to report with decisions and if require then make available the facilities of experts/specialists. The design of this architecture provides the abstraction of multiple layers that can increases the level of abstraction and can check the performance and test the validity / output of each layer. This style is well suited in the case when patients are clearly elucidate all the symptoms of diseases and these should be matched with the signs of the diseases, if, signs are moderately matched then this style is not well suited and conditions need major support to the external systems. The external systems augment results in additional cost and efforts.

2.2 Separated Presentation Architectural Style (Model View Controller)

The foremost advantage of the decomposition technique is to evidently separate the components and their functions in user interface, data storage and processing logic. [23]

The figure No.3 demonstrates the separated presentation architecture styles use MVC style which specifies the three separate entities; model, view and controller. This architecture style when used for medical domain, consider two views i.e. doctor and patient view. The model stores the medical history of patients at different instances of time. In medical architecture of MVC the controller is regulator. The controller regulator specifies that the medical events are performed either by doctors, supporting staff, technical staff etc. with the help of medical equipments or devices and laboratory tests.

The model contains the history of signs of diseases according to symptoms of patients. It encapsulates the objects according to medication with proper medical constraints. The doctor's view focused on the patients' health condition after medication and counsel to patient while the patient's view center of attention is on growth in fitness and health check. The regulator controller supervises the patients' growth on the basis of patients' current condition. It selects the strategic patterns of medication according to the signs emerge. It also notes down the behavior/health of patient during diagnosis and store this information in encapsulated form into model. Due to its basic

nature, it is reusable and can perform test on each components with effortless protection. This style is well suited in the case when patients does not much aware about the diseases and cannot enlighten the signs of diseases. The patient can only describe behavior of the actions occurs with himself. The regulator regulates the signs with different orientation, identify the diseases, indispensable causes and effect of the diseases, symptoms of the diseases and well proper strategic planning doctors try to cure the patients

2.3 Communication architecture style (Pipe and filter Architecture Style)

This architecture is most suitable when input data are communicated through a series of computational or executed components into output data. The pipe and filter is an example of communication architecture style. A system executes the pipe and filter architecture achieves desired output by using a number of smaller transformations and integrate them in a form which is overall desired output of the architecture. The architecture contains two entities; Pipe and Filter. Pipe works as connector between input data & filters and filters & output data. The process may have number of pipes and filters. This style contains some constraints are as follows:

1) The filter does not know the identity of the source which provides the data to the filter. The filter knows only the data elements which it requires.
2) A pipe is bi-directional connector i.e. it provides the connection for input which passes to the filter and other connector for receives the output from the filter after computation or manipulations.
3) The filters are independent from the other filters i.e. each filter is not dependent on the others.
4) Each filter has own independent mechanisms to perform the computation.
5) There is no interaction between the filters.
6) No feedback in architecture i.e. once pipe moves from one filter to another than no provision to come back on the previous filters.

The components architecture is reuse because each file is responsible for specific operations. Even, the combinations of any two filters are possible. It can transmit the information and reuse in other application. The main characteristics of this architecture style is that it can be easily maintained and extend. New filter can be added and old filters can be easily replaced with advance filters. When the architecture implemented in sequential manner then there is no deadlock occur in any operations. This architecture support concurrent execution i.e. filters can execute in parallel manner [25].

The figure no. 4 is the architecture style when used in medical domain. This consists of four filters and one pipe. The pipe provides the information of patients' to the essential filters. Initially, the physical resource filter preprocesses the data. Secondly, Data preprocessor filter recognize the signs and symptoms of specific patients and passes to medication for computes the information taken from the outcome of the physical resource filter with required step of diagnosis. The Treatment filter will produce the report of patients and provides the interaction with the external system, if required. The presentation filter will make inquiries concerning the status of the patients through each session. This style is well appropriate in the case when patients are well attentive of symptoms and can provide inclusive set at the time of analysis. The specialist knows the methodical evaluation of the symptoms with reference to the diseases signs which can be filtered and can adjoin the additional filters from external sources (if required), in order to reduce the overall impact.

3. PROPOSED ALGORITHM 1: For choosing Architecture style

This algorithm is the bases for choosing appropriate architecture style. The significance of notations is explained in algorithm. The paper accentuate on three different architecture styles namely; layered architecture style, Separated Presentation Architectural Style (model view controller) and Communication architecture style (Pipe and filter architecture style).

Step 1: Construct a signs and symptoms set Δ_S over U
Where U: universal set, Δ_S: Set of Signs and Symptoms

Step 2: Assign appropriate weights to sign with respect to normal range of sign.

Step 3: Find the aggregate set Δ_x* of Δx by the following definition

$$\mu_{\Delta}(U) = \frac{1}{|p|}\sum_{s\in p} \mu_q(\mathrm{S})\, \mu_{s_i}(S)(U)$$

Where $\frac{1}{|p|}$ = Cardinality of p or S and $\mu_q(\mathrm{S})$ is set of q^{th} sign and $\mu_{s_i}(S)(U)$ is the set of parameters in the patient's sign in the universal set.

Step 4: Find the largest aggregate value S μ_{Δ_S}* (U) from step 3.

Step 5: Perform the analysis of step 3

5.1 If |S| = |1| set having only one test sign as max aggregated weight values then apply communication (pipe and Filter) architecture style. (As per algorithm no. 2)

5.2 If |S| > |1| and Δ_S {|S_j - S_i|} < e, Threshold error then apply layered architecture style (As per algorithm no. 3)

5.3 If |S| > |1| AND {S} → {S} × {U} then apply Separate Presentation architecture style (model view controller) (As per algorithm no. 4)

5.4 If |S| = |1| and {S} → {S} × {U} then apply communication (pipe and Filter) architecture style.

Algorithm 2: Layered Architectural Style Algorithm

Step1: Repeat the step 2 to 6, till the signs value are normalizes .

Step2: Extract the datafrom the physical resource layer by means of the data access components at data preprocess layer with the proper data selection through data selection component and if required do the data preprocessing through the data preprocessor components.

Step3: Repeat step No. 2 whenever data access component gets indication from physical resource layer.

Step4: The resultant of data preprocess layer is analysed with medication constraints and medical test through analysis of sign and symptoms component, Medication constraints components and medical test components respectively, which will inclined to normalized sign values.

Step5: From medication layer the resultant are transfer to session layer where reports and recommendation interface get activated. If report and recommendation compels (signs values are not as per the medication output) than take the external recommendation and go to step 4 for the medication process.

Step6: As the patient is well aware, hence the interaction of patient must be with each layer.

Algorithm 3: Separated Presentation Architectural Style Algorithm

Step1: Prepare the database -- on the symptoms (data) provided by the patients.

Step2: Regulator (may be devices also) supervise the patient with the appropriate selection of the patterns which is previously evaluated by algorithm no.1.

Step3: Monitor the model behavior and results are conveyed to both patients as well as experts (Patients' feedback is must in this style to get more information)

Step4: Center of attention is the models actions (in terms of the sign values normalization) and models encapsulate the information with signs and constraints of the diseases object.

Step5: Repeat step No. 1 to 4 till the signs values comes into normal range or no more information provided by the patients.

Algorithm 4: Pipe and Filter Algorithm

Step1: After hit upon the major sign the expert start the assessment in the form of filtration and repeat step 2 - 5 till the sign value comes within range with + / - threshold value.

Step2: Partitioning the patient on the basis of sign and symptoms, attending the older of medication.

Step3: Communication is required to coordinate the stages considering communication structures and procedures to define the pipe components as sign of the disease.

Step4: The pipe and filter structure defined in the steps 2 & 3, should be evaluated with respect to the symptoms and the physical circumstances of the patient and its cost with respect to health initially.

Step5: Each sign or symptoms with current circumstances are assigned to a medication attempting to maximize the patients' health condition with minimizing the after effect and cost on the patient health situation. Hence we can say that optimizing the patient state.

4. A CASE STUDY (Experimental work)

The experimental work is carried out on liver data set which contains 583 patients from government hospital with parameters are Total Bilirubin, Direct Bilirubin, Alkphos Alkaline Phosphotase, Sgpt Alamine Aminotransferase, Sgot Aspartate Aminotransferase, Total Proteins, ALB Albumin, A/G Ratio Albumin and Globulin Ratio. Their normal range of values is as follows: 5.1 – 17, 1.7 - 5.1, 44 – 147, <=45, 14 – 20, 6 - 8.3, 3.5 – 5 and 1 - 2.5 respective.

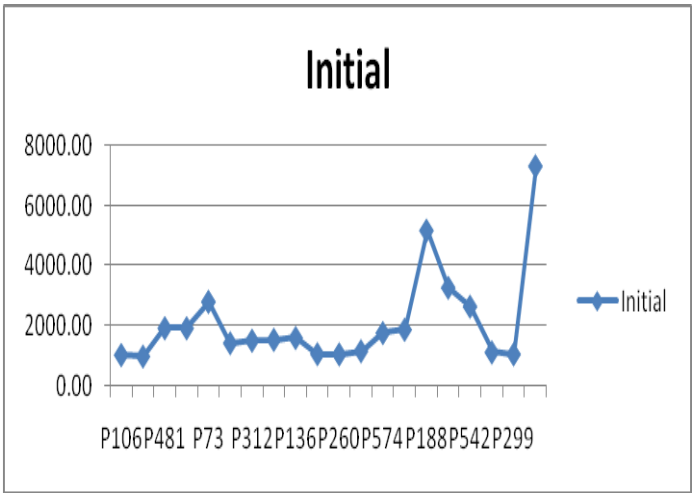
The Graph no.1 shows the severity level of patients' and results that patient P299 is the most severe than others. This graph is designed trivially on the patients' statistics comparison without applying any algorithm with no association of the architecture styles, i.e. no investigation on the patients' pattern. Thus, this represents that at first instance by reviewing the parametric values the controllers (doctors / experts) can predict the patient to be concentrated first.

The Graph no.2 shows that patient no. 106 is the most severe than others after apply the algorithm no. 1 and calculate the weight of parameters with corresponding to normal values. Though, graph no.3 shows that the comparison between the severity of patient with initial value of parameters and filter values of parameters and it is clear that patient no. 106 is the most critical patient.

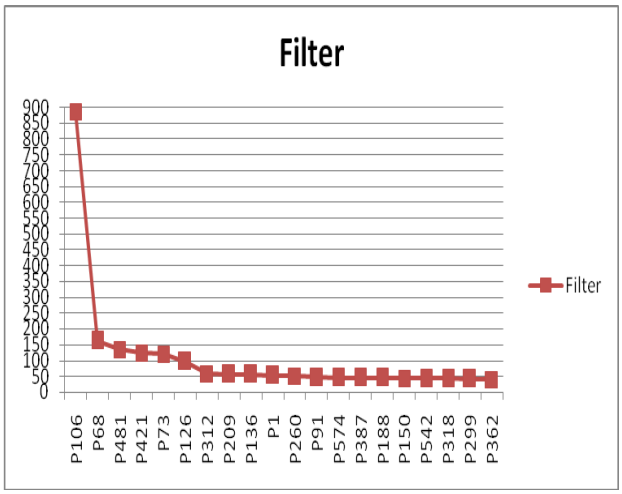
The proposed algorithm results in four different scenarios as (i) patients well aware to (ii) either partially aware or ignorant (iii) medically well aware and (iv) aware but affected with other signs. As per the figure no.5 the selections of the architecture style based on the above scenario are as follows {conditions 1-4}.

The layer architecture is more suited for the scenario no. 4 which can be easily verified and demonstrated in graph no.4.

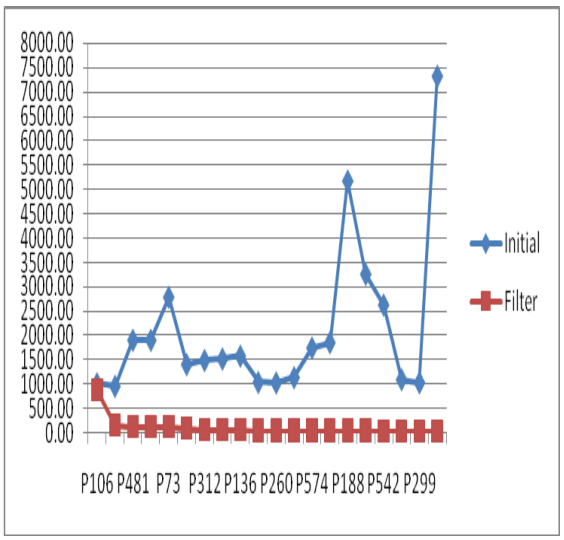
The graph No. 4 (layered) illustrates that the patient p106 (consider the serve patient as per proposed algorithm 1 - step 4) under goes 9 sessions for the normalized sign values. Here 0 indicates the normal range (normalized values) of the signs. Gradually the values are processed layer wise to inclined on the way of the normal range



Graph 1 initial values of signs

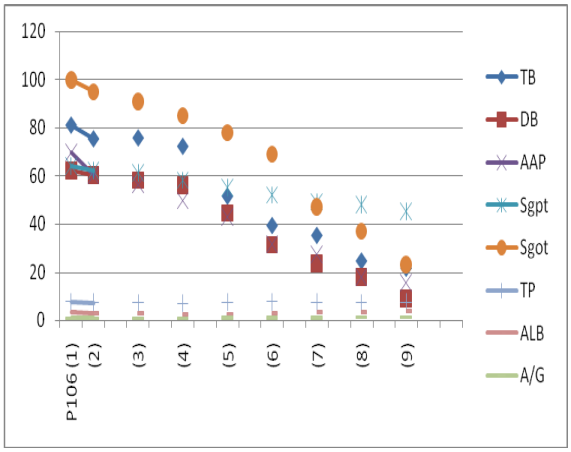


Graph 2 Filtered values of signs

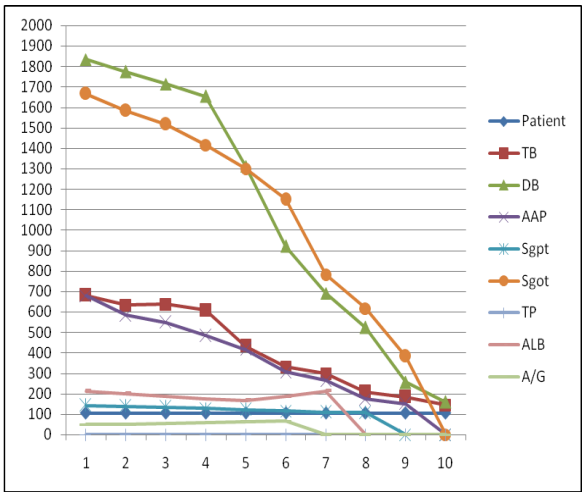


Graph 3 Comparison between initial and filtered values of sign

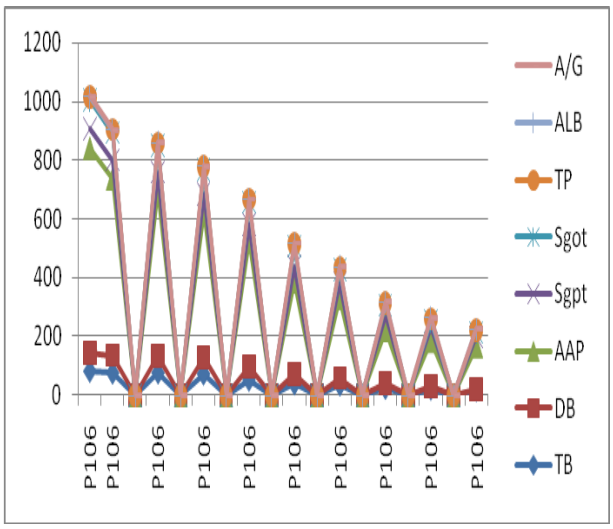
4. ANALYSIS



Graph 4 layered Architecture on over all Symptoms



Graph 5 MVC Architecture on over all Symptoms

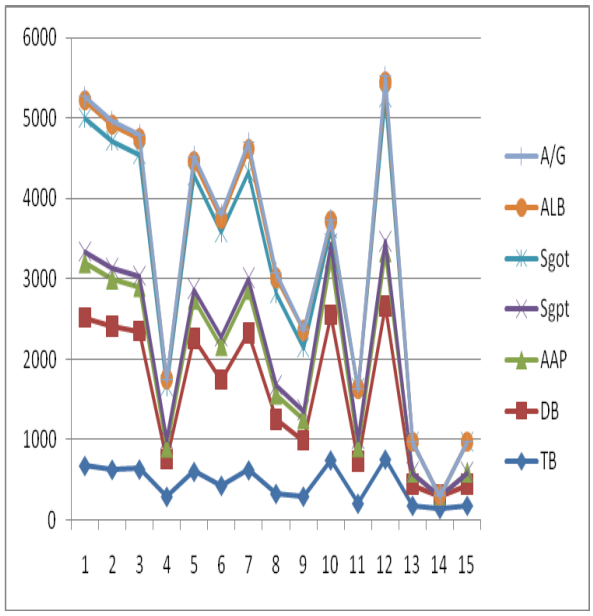


Graph 6 Pipe and filter Architecture on individual symptoms

For Scenario no.2 MVC is more suited which can be verified in the graph no.5. In this style the database is prepared as per the patients' feedback in every session and transformation of the signs with the influence of other signs due to the encapsulation and constraints. After 10 sessions patients attain the normalized values.

For Scenario no.1 and 3 Pipe and filter is more suited which can be verified in the graph no.6.

In this style the values of the signs are filtered in every session. As per the graph no. 6 the sign values are filtered with the integration of medication based on symptoms. As there is no feedback process, the filtration process was based on medical tested. After 10 sessions patients attains the normalized values.



Graph 7 MVC Architecture with external systems

If the appropriate architecture style were incorrectly selected, it results in wrong medication thus elongate the recovery process and at times may cause danger to the life also. The fact is well demonstrates in the graph no. 7 where one has to apply the MVC, but pipe and filter is preferred and the result is self explanatory, i.e., the sign values are in decline position but as no feedback is in pipe and filter the filtration process hold on with wrong medication which will not only increase the duration of

the recovery process as well as risk to the life also with after affect of wrong medication.

5. CONCLUSION

The paper well describes the meta-model of medical software with the help of Software architecture styles. This model represents the basis for the software development in order to fasten the diagnosis as well as the treatment process even in the absence of the experts' hand. The outcomes of the proposed model conclude the importance to classify the patients' by weights assignments based on the signs and the symptoms. It also spotlights on the selection of software architecture styles that are useful in different classes of patients with level of the patient's severity. Thus, this concludes that if single disease sign with single priority sign use the pipe and filter architecture styles, if single disease sign with multiple priority signs layered architecture styles is to be used, if multi disease sign with multi priority sign use the model view controller architecture styles and with the multiple disease sign and single sign the use of pipe and filter architecture styles is beneficial. These recommendations are tested on the liver data sets. The model focused to fasten the recovery process but with the limitation of cost effectiveness calculations.

6. REFERENCES

[1] Prakash M. Nadkarni, Randolph A. Miller, "Service-oriented Architecture in Medical Software: Promises and Perils", Journal of the American Medical Informatics Association, 2007

[2] http://en.wikipedia.org/wiki/Program_architecture

[3] http://hercules.gcsu.edu/~adahanay/mmis6298/softwarearchitecture/chapter02.ppt

[4] Bass Clements, and Kazman, Software Architecture in Practice 2nd Ed, Addison-Wesley 2003.

[5] http://www.scribd.com/doc/113046308/software-architecture-for-developers-sample-pdf

[6] Abdel Ejnioui, Mathieu Morjaret and Carlos E. Otero, "Software Architecture and Prototype for Supporting Medical Prescription Adherence Using SMS Services", Computer Technology and Application 3 (2012) 616-623, David publishing.

[7] http://leanpub.com/software-architecture-for-developers/read

[8] Lori A. Clarke, George S. Avrunin, Leon J. Osterweil, "Using Software Engineering Technology to Improve the Quality of Medical Processes", ICSE'08, May 10–18, 2008, Leipzig, Germany.

[9] http://www.sei.cmu.edu/library/abstracts/news-at-sei/architectdec98.cfm

[10] Wang, "Modelling information architecture for the organization. Information and Management" 32, 6, 1997, pp. 303-315.

[11] R. Weber, "Conceptual modeling and ontology: Possibilities and pitfalls", Journal of Database Management,14, 2, 2003, pp. 1-20.

[12] P. Nykanen and J. Makinem, "Integration of medication information in electronics patient record systems", The dementia patient case. Turku School of economics, Research report LTH-1:2007, Turku, 2007.

[13] Lim, W. C., "Effects of Reuse on Quality, Productivity and Economics", IEEE Software, Sept.1994, 23-30.

[14] Makinen,J. Et.Al., "Process Models of medication Information", Procd. Of the 42nd Hawaii International Conf. on System sciences 2009, 1-7.

[15] Clark, L.A., et. Al., "Using software Engineering Technology to improve the quality of Medical Processes", ICSE'08,May, 10-18, Germany.

[16] M. Champion, C. Ferris, E. Newcomer, and D. Orchard, "Web Service Architecture", W3C Working Draft, 2002. http://www.w3.org/TR/2002/WD-ws-arch-20021114

[17] Hamed Yaghoubi Shahir, Ehsan Kouroshfar, Raman Ramsin, "Using Design Patterns for Refactoring Real-World Models", 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009, pp 436-441.

[18] U.Banodha , K.saxena, "Impact of Pipe and Filter Style on Medical Process Re-engineering", International Journal of Engineering Sciences, October 2011.

[19] U.Banodha , K.saxena,"Usability of Software Architecture design pattern in Medical process reengineering model", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 6, June 2013, ISSN 2319 – 4847.

[20] U.Banodha, K.Saxena, "Comparison of Software Architecture Styles in Medical Process Re-engineering Model", International Journal of Wisdom Based Computing, Vol. 2(1), April 2012.

[21] U.Banodha, K.Saxena, "Dynamic Software Architecture for Medical Domain Using Pop Counts", International Journal of Advanced Computer Science and Applications, Vol. 5, No.2, 2014

[22] Garlan and Shaw (January 1994, CMU-CS-94-166, http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf)

[23] Mahesh P. Matha, "object-oriented analysis and design using Uml", PHI learning Pvt. Ltd., 2010

[24] Bass, Clements and Kazman, Software Architecture in Practice 2nd Ed, Addison-Wesley 2003

[25] www.coronet.iicm.edu/sa/scripts/lesson08.doc

7. AUTHOR'S PROFILE

Umesh Banodha, Assistant Professor at Samrat Ashok Technological Institute, VIDISHA (M.P.), an Autonomous Institute, affiliated to Rajiv Gandhi Technical University, Bhopal. I did MCA, M.Tech (Honors) and Pursuing Ph.D. My Area of interest Software Engineering / Architecture, Databases, UML, object oriented, Programming Languages etc. I am member of various international / National journals. I published more than 12 research papers in various conferences and journals (National / International).

Kanak Saxena, Ph. D. in computer Science from the Devi Ahilya University, Indore, INDIA. She is professor in the Computer Applications Department at the Samrat Ashok Technological Institute affiliated to Rajiv Gandhi Technical University, Bhopal. Her Current research focuses on Database Systems, Parallel computing, Data Uncertainty and design and other interests include Network security and performance and Software Engineering. She is the member of editorial board of

various international journals. She is the member of the international committee of the International Conference on Computer Science and Its Applications. She Published more than 80 research Papers in Various Conferences and Journals National / International).

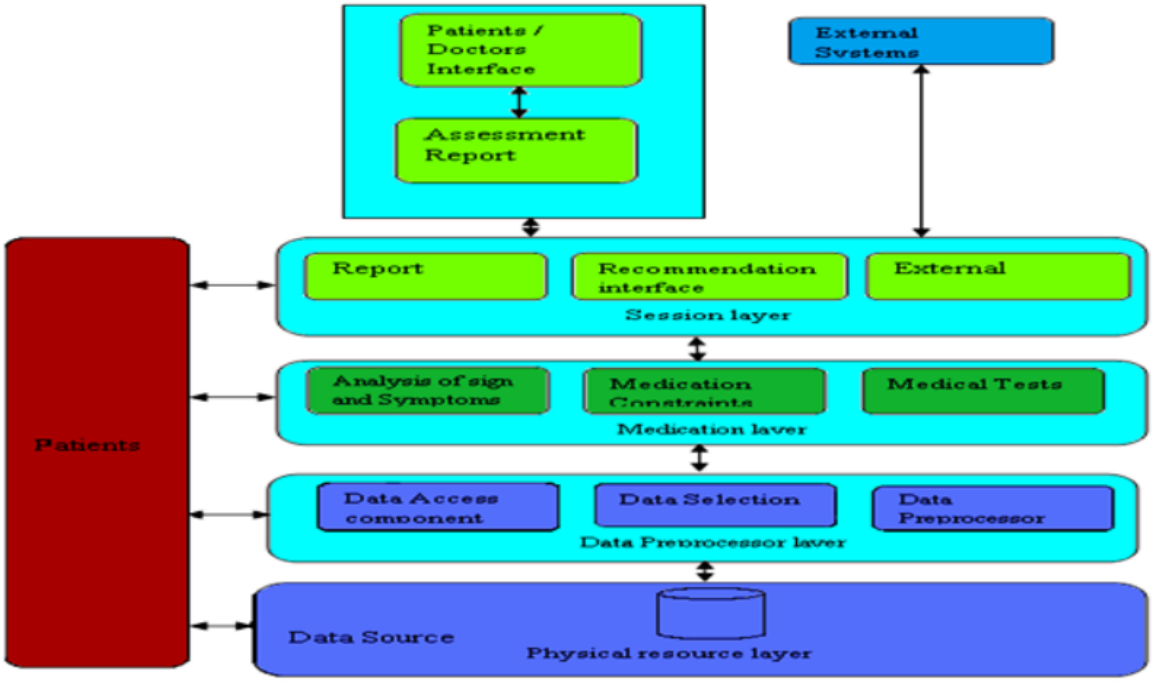


Fig 2 Layer Architecture of medical domain

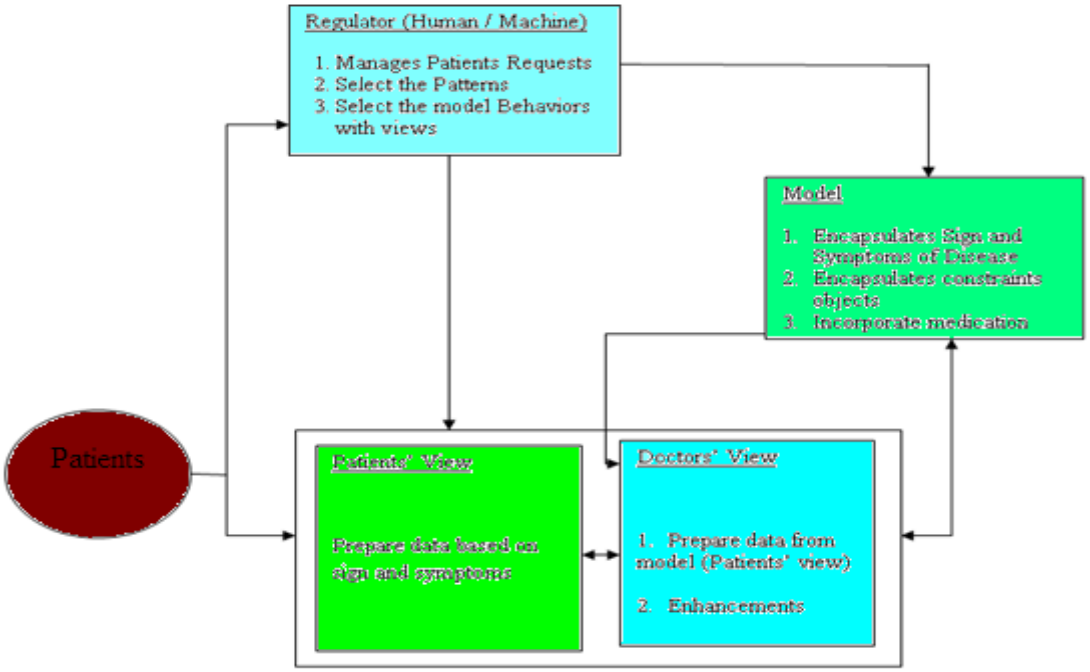


Fig 3 Model View Controller Architecture of medical domain

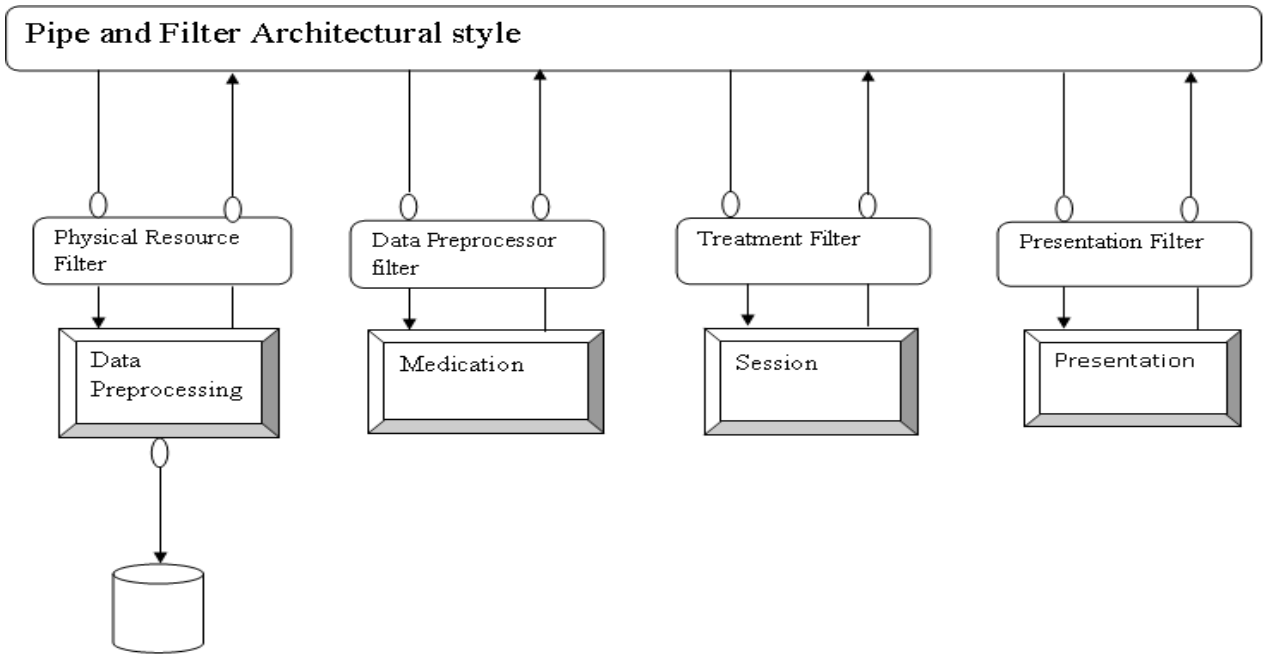


Fig 4 Pipe and Filter Architecture of medical domain

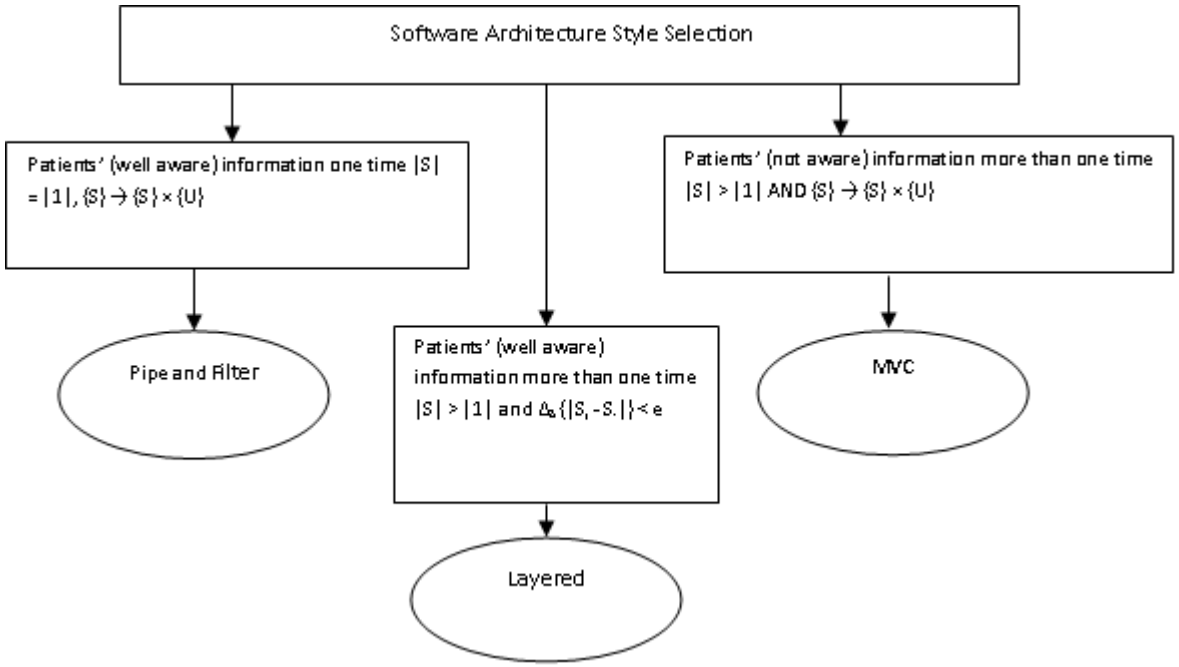


Fig 5 Software Architecture Style Selection Process