Volume 5, Issue 5, May 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Tailorable Software Architecture in presence of Non-Functional Requirements

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

Abstract — Medication is one of the most important areas for society and human beings. This area is most dynamic, flexible, uncertain and highly affected with the environment so that it is very complex and challenging to develop the framework architecture. The very challenging task is how one can develop such software architecture. The study has shown interests of designer, developer, architects, analyzers and practitioners in using software architecture towards the development of a flexible system using quality attributes. The software architecture is high-level generalization of the context, problem, solutions, resulting context, related requirements and most important in design decisions. This paper exhibit MSDS (Medical software development system), a frame work model which support the development of tailorable software architecture using various Non-functional quality attributes. The purpose of MSDS is to act as the basis for creating suitable architecture systems for medical domain with improvements and reusability to show the structure of an inventive model.

Key Words — Non-functional requirements, Tailorable software architecture, Medical software development system, Quality attributes, UML Diagrams.

I. INTRODUCTION

Once a new technology rolls over you, if you're not part of the steamroller, you're part of the road.
–Stewart Brand

There is a high demand of quality in the health care sector. This demand has grown up after the creation and development of computer and software products. As the era grows current approaches concentrate on algorithmic augmentation, but mostly either fail or inaccurate in medical domain. Thus, there is the necessity of robust software architecture which must have grip in a medical domain.

The object of software architecture is to capture the persistent parts of the program and to derive the transient versions using the architecture refinement. The software architecture is an abstraction of the run-time elements of a software system during some phase of its operation. A system may be composed of many levels of abstraction and many phases of operation, each with its own software architecture [1, 2, 3, 4].

For this, the professionals diverted their interest in using flexible software architecture towards the development of a tailorable system which makes it easy and even interesting. Architecture is well known solutions for reoccurring problems. By classifying software architecture, a list of components can be created for use when building tailorable complex software systems for the medical domain. They can reduce augmentation time as known solutions as are tried and tested. Thus, tailorable software architecture improves the quality of the architecture and their components [20].

This paper presents MSDS (Medical Software Development System) - a framework which is proposed to support the development of tailorable software architectures using components in the light of NFR (non-functional requirements). The Framework is to design in which software tailorability to be achieved if not absolutely but within acceptable limits. NFRs are important in any real-time models but are difficult to express due to the following reasons:

1. Certain constraints are related to the design solution that is unknown at the requirement stage.
2. Certain constraints are highly subjective and can only be determined through complex empirical evaluation.
3. NFR tend to be related to one or more functional requirements.
4. NFR tend to conflict and contradict.
5. No universal rules and guidelines for determining when non-functional requirements are optionally met.[5, 19,21, 22, 23, 25, 26]

The development of the MSDS is on the aim of tailorable software. In MSDS, the software tailorability as a NFR is treated as a sub-goal which is to satisfied if not absolutely but accepted within acceptable limits. The major problem is with the tailorable requirements is, it's potentially huge if rather not infinite but somewhere the conflicting NFRs. The main focus is to explores both definitional and design alternatives to satisfies the tailorable sub-goal, analyzes tradeoffs among the alternatives, estimates the impact of the alternative components and quality attributes upon software tailorability and selects among the alternatives in the relation to the medical characteristics.

The architectural design process using MSDS would proceed as per the following guidelines.

1. Post tailorability requirements along with any other important NFRs as well as functional requirements.
2. Refinements of the NFRs and prioritize them, taking into consideration any particular characteristics of the medical domain.
3. Considering the architectural concepts and alternatives at a worldwide scope to meet the both functional and nonfunctional requirements stated.
4. Considering the components in a way which will be based on the architectural concepts and alternatives being considered with the focus on the context and problems associated with the each of the analysis phase.
5. Consider the quality attributes in a way which will be based on the architectural requirements and improve the output [20, 27].

II. MEDICAL SOFTWARE DEVELOPMENT SYSTEM (MSDS)

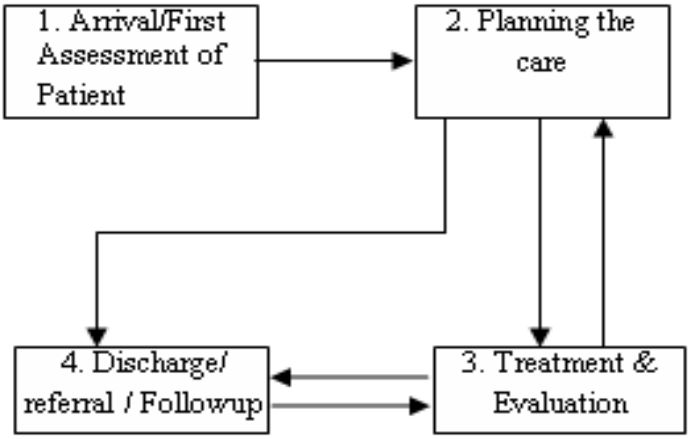

Figure 1 Medical Process Modeling Approach

The figure 1 characterizes the medical process modeling approach which exemplifies the inclusive progression of the patient. The approach is classified in four phases as (i) Arrival/First Assessment of patient, (ii) Planning the care, (iii) Treatment & Evaluation and (iv) Discharge/Referral/Follow up. MSDS conceptually based on the second (planning the care) and third phase (Treatment and evaluation) of Medical process model as of figure 1. The motivation for the development of this process model was the identified problems in accessibility and usability of all relevant medication information in medical decision making and treatment follow up. The problem in accessibility and usability of information are related to the coverage of medication information which is very flexible in nature if not static [8, 9, 10, 11, 12, 13, 14, 15].

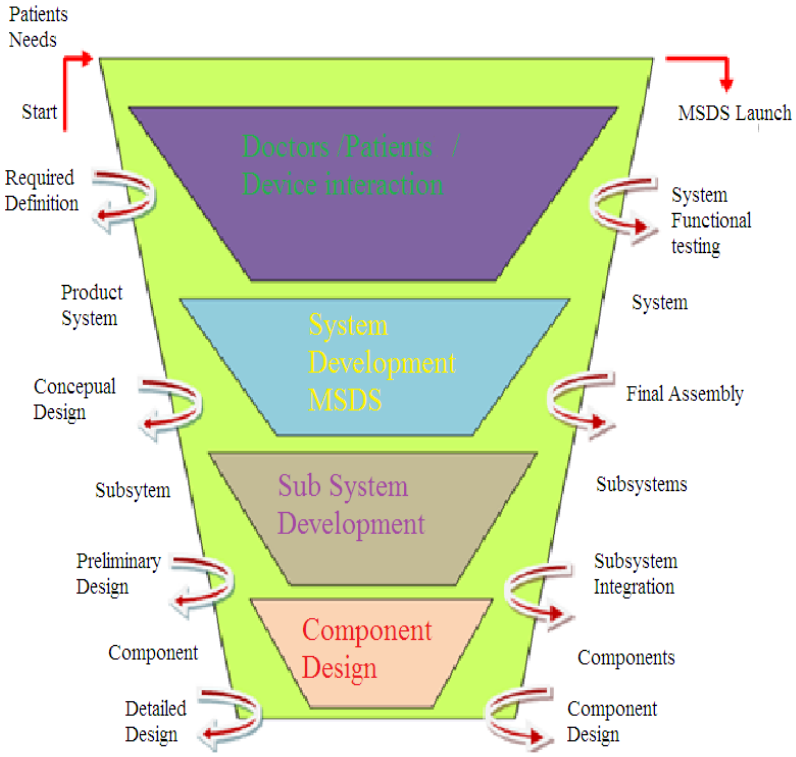

Figure 2 Medical Software Development System

The figure 2 exhibits the process of Medical Software Development System which initiated with the patients needs and output with the MSDS launch. The MSDS is based on decomposition modeling approach; the central part of the diagram 2 shows the step by step decomposition of the MSDS part. The requirements of left side of diagram are inputs as patients need (start) with interface to MSDS components from top to bottom while the output and interaction of right side is produced by various components of the MSDS from bottom to top.

The MSDS activities start when patients arrive. He provides / describes complete sign and symptoms of diseases, which required the interaction with doctors and devices. On the basis of complete and detailed information (required definition) MSDS developed the system, which is a product system which further decomposes into subsystem on the requirement of the core and conceptual design. Further, MSDS produced the independent components on the preliminary requirements with the detailed design in presence of the non functional requirements between various components of MSDS. Integration is required to collect the independent component and produced the subsystem, and with time all the subsystems are integrated and produced the system. At the last MSDS summed up with testability applied on functional requirement of the system and Launch MSDS. In order to instigate MSDS as tailorable, the tailorability is decomposed into two descendents subgoals as per the figure 3.

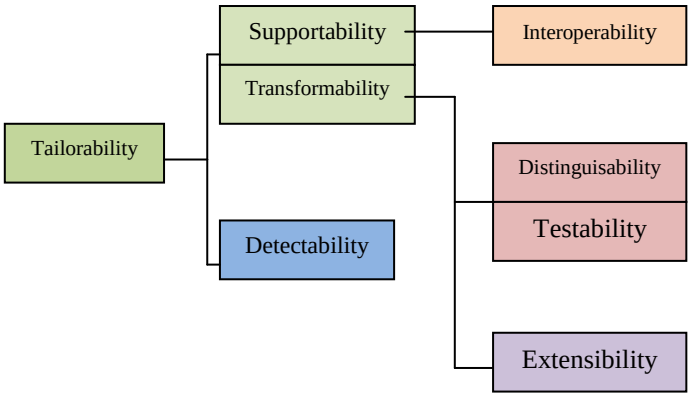

Figure 3 Hierarchy of quality attributes in MSDS

In perspective to NFR, the first step is the Tailorability. In medical domain the first step towards developing the traliorable MSDS is refinements of flexible requirement. The term tailorability means focus on the Flexible architecture (rule-based and component based architectures with simple or compound Components) with appropriate interfaces (visualizing and manipulated tailored artifacts; 2D & 3D interfaces. Describing tailored artifacts. Understanding tailored artifacts, providing support for tailoring i.e. integrating checking and accessing the tailoring functions i.e. direct activation).

Detectability (change in medical environment) and transformality and supportability (support in MSDS may be reusable). Thus to obtain the tailorability one should satisfies both of its descendant supple goals. Maximize the capacity to detect changes in the medical environment and maximize the capacity to transform the system with the support of the medical equipments or software. Further, the transformability is decomposed into two supple objects; distinguishability (change in MSDS) and testability (change in MSDS) respectively for maximizing the capacity to distinguish the needed change to be made in the system (with respect to phase 3, figure 1) and maximizing the capacity to make the changes with help of verification.

A. MSDS Activities

1) Requirement Analysis

The requirements are fulfill by the existing components or not. The important and careful task is to negotiate with required symptoms and modify them to be able to use existing components. These components are integrated, i.e., the integration process includes integration of standard rule-based components that build a component framework and the application framework, thus in the resultant of a sub system [16].

The final phase which is the resultant of component design, subsystem and system development is the patient/doctors and device interaction which is the key issue in MSDS. In the medical domain, its recognized that the architecture is not static as its nature is flexible due to the following properties : (i) *continuing change*: a large system changes undergoes continuing change or becomes progressively less useful either from the climatic change resultant in the changes in routine performance or due to the presence of new viruses etc. (ii) *Increasing complexity*: diagnosis and treatment becomes more complex, thus software evolves its complexity increases unless treatment is done to maintain or reduce it and the last is its evaluation which is very significant and individuals' act in response on medicines.

Requirement Analysis: Means what the MSDS supposed to do it. As per figure 4, the emphasis on

(a) requirement gathering, i.e., patient and doctors are to determine and satisfies on the requirements for the diagnosis and treatment and evaluation.

(b) Analying requirements: determining whether the situation/scenarios of requirements in medical domain are unclear, incomplete, ambiguous or contradictory and then resolving these issues with existing components or need of new development. If the component exists then proper selection must be done with adaptation and testing.

(c) For recording and further processing one can use the standard tools of NLP, use cases, process specification, OCL etc.

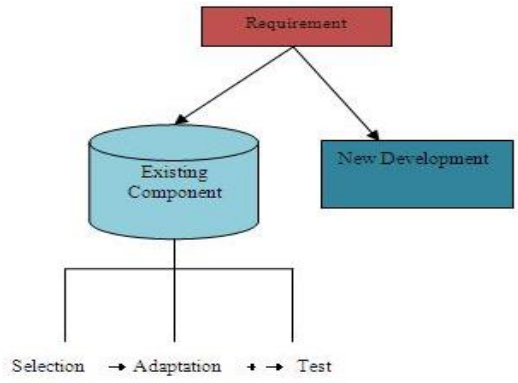

Figure 4 Requirement Design

2) Component Design

Components are built to be used or reused in many applications, some possibly not yet existing in some perhaps unforeseen way. In MSDS, component design focused on the identification of reusable entities and relationships between them that begin from the system requirements and the availability of components already exists [6, 7, 24]. The foremost problem is in the selection of the components as one cannot distinguish among similar symptoms in two different diseases or dissimilar symptoms in two comparable diseases. Thus, selection is a significant task followed by integration and testing of components. If the components are not obtainable then there must be some provision for development of the new component.

3) Conceptual design and primarily design

An overall architecture on one side we have focused on the tailorability of the MSDS but on the other hand in the use of emergency, some actions to be performed; thus one more supple object is also added as prompt abilities which are of paramount importance.

4) Detailed design

MSDS units are integrated. The formalization of the detailed design in an executable way that can be composes of smaller units. The non functional requirements are recommended in each phase of the MSDS. Tailorable software architecture as shown in figure 5 specifies the areas where the NFR are extremely essential. The detectability is vital to identify and predict the patient diseases(s) and also required the appropriate diagnosis selection. Once identified and categories the patients need transformability, which is indispensable. In this NFR the doctors can take some measures like references from medical experts, medication and various medical laboratory tests are mandatory. The distinguisability is recommended in action of comments provided by medical experts / Specialist doctors, medical signs, symptoms and prediction of diseases(s) and medication also. The term testability is essential in medical laboratory test and evaluation of patient status after the defined time span. If patients' improvement then it will reproduce in proper medication with time span else recommended for new or modify medication with new medical tests.

The term extensibility is recommended when amendment of the medication at different stages have to prepare. The terms supportability and interoperability is promising in evaluation of medication direction based on the patients historical record, doctor's suggestions, precautions with progress and feedback by hospital management system.

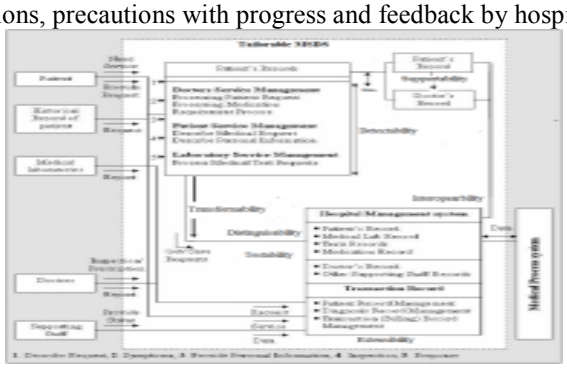

Figure 5 Tailorable Software Architecture using Non functional Requirements

III. ANALYSIS

A model is the presentation of complex reality in ease and simplified way, which means to create abstraction. It allows eliminating unwanted details and focusing on all the details at various instance of time. The efficient UML model also express the discussion among various stakeholders of the model, Concentrates on key fundamental and also focuses on work to achieve the common goal of various stakeholders. An efficient UML model can support to create other

models also which express the different view of the world. Modeling (e.g., with UML) has been accepted and established as a means of analyzing and designing software. In order to create the MSDS, the domain model in which the software systems operate must also be model, understood, and sometimes improved. The domain model is the center for conducting MSDS or improving how the MSDS is operated. The evolving models also help the developers' structure and focus on their opinion. Working with the models increases their understanding of the medication and, hopefully, also their awareness of new opportunities for improving medication and invention of latest medication for new diseases.

There can be many reasons for adapting UML modeling:

1. The models can be used to provide a clear picture of their role and tasks in the overall tailored architecture.
2. To act as the basis for creating suitable information systems that supports the tailored architecture.
3. To act as the basis for improving the tailored architecture and their operation.
4. To show the structure of an innovated MSDS. The model becomes the basis for the action plan.
5. To experiment with a new tailored architecture concept or study a concept used in a competitive environment. [17, 18]

A. Activity Diagram

The important UML diagram for MSDS is the activity diagram. It is supported by various UML tools. Activity diagram are useful to describe the activities of software architecture and specific operations which are implemented on this tailorable architecture. Activity diagram show the architecture from various point of view like sequential or parallel execution of operations, object used by activity, role and responsibility of the activity and interoperation between activities.

The Activity diagram shows the input and output of MSDS. A MSDS is a collection of set of various activities. The MSDS set of activities produced specific output of an individual patient. It also describe that how work is performed in MSDS organization for betterment of the patient. The MSDS specifies the sequence of activities along with time and place. It also specifies the beginning, end, and an input & output of activities and architecture for implementation. It also expresses the activities which take information from different resources or these are mandatory activities like Doctor/ device interaction, system/ sub-system development, component development and non-functional requirements for MSDS. These activities are helpful to produced traliorable MSDS in perspective of quality attributes.

B. Medical Process Diagram

The medical process view is the core part of this tailorable software architecture. Figure 7 shows a medical process diagram where the relationships between different processes are exhibit. The processes portray the different activities. Whenever there is a need of workflow of medical process, to partition the activities into group. Each group is a swimlane. Swimlane is a locus of activities. Each group represents the tailorable software architecture responsible for these activities. There is a relationship between processes and activities with the resources available during the interaction to achieve the desired goal. Resources of tailorable software architecture are patients, doctors, devices, supporting staff etc. the diagram show that how medical events are generated during different group.

The activity of group1 is problem definition, which provide the patient complete information (personal and medical) and process of this group classifies provided information and convert it into appropriate sign and symptoms.

The activities of group2 are doctors' inspection or take the services from experts / specialized doctors. These resource works under certain circumstances within controlled manner. The process of this group generates prescription, direction of medication, what should be the expected diagnosis of the patients. The resultant may be depending upon the activities of group3 if medical laboratory test are essential. Thus, these events are transition and may cross Swimlane.

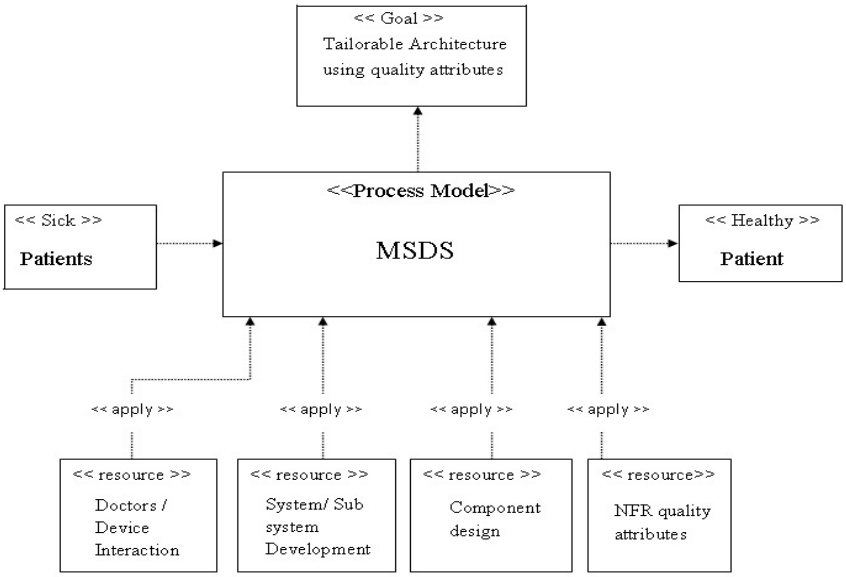

Figure 6 An Activity diagram for MSDS

The activities of group 3 are to take advantages of other supporting staff during proper treatment and evaluation of patient and provide the medical laboratory tests. The process of this group communicates with doctors. The doctors provide the proper medication and diagnosis to the patient. Doctors also evaluate the response of medication over the patient and it should decide the direction of the medication. The activities of this group are repetitive in nature. If no improvement in the patient then doctor again perform the same process until not recognize the exact problem and provide the medication and diagnosis.

The activity of this last group is associated with transaction section of the patient. If patients are satisfactory fit then only the patient discharge. The process of this group deals with the financial assessment of the patient. If all activities and process performed then patient is discharge. Finally, a healthy patient is the output this medical process model.

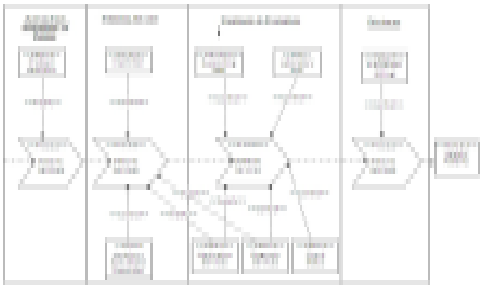

Figure 7 A Medical process diagram based on an activity diagram with Swimlanes

C. MSDS Vision view

The MSDS vision view expresses the goal of tailorable software architecture. It evidently specifies how to achieve the goal, what and where the problems can arise. The figure 8 is goal problem diagram based on UML object diagram. This vision view specifies that how the problem is decomposed into various subgroup or component. It also specifies that where the roles of non functional requirement with various components occur with its implementations (if required) to fulfill the goal. Finally, all subgroups are integrated in proper order to achieve goal.

In this, the goal is to achieve the patient (healthy). This problem is decomposed into production system, production sub-system and component system. Further, it decomposed into lower level activities like patient, doctors and laboratory sub system and detailed design.

This is the actual detailed area where real world object is created and non functional requirements are implemented. For this one has to produce the object of real entities and all objects are qualitative goal of vision view. Finally, integrate the entire design component with next upper level subsystem and continue till final goal is achieved.

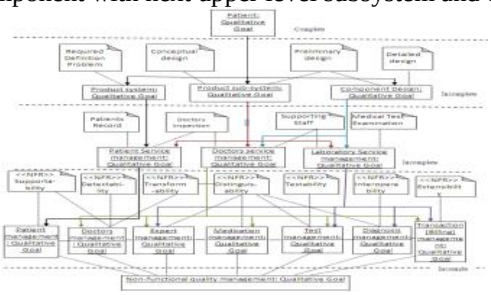

Figure 8 A goal-problem diagram based on a UML object diagram

IV. CONCLUSIONS

The proposed MSDS- a framework which is intended to support the development of tailorable Software Architecture in presence of NFR quality attributes for medical domain. The paper is a roadmap for developers, practitioners, systems architects and analysts. The analysis is done using UML with an approach to way of (i) Identify the information systems that best support the operation of the medical domain, (ii) analyzing non functional requirements and selection of the components. These components are decomposed in hierarchy with interfaces. MSDS further enhanced with the UML activity diagram, Medical Process Diagram and MSDS Vision view.

REFERENCES

[1] www.stackoverflow.com/questions/704855/software-design-vs-software-architecture
[2] www.jatit.org/research/grid-software-architecture6.htm
[3] www.hta-inc.com/Download/StandardSoftwareArch.pdf
[4] http://roy.gbiv.com/untangled/page/2
[5] http://www.iai.uni-bonn.de/III/lehre/vorlesungen/SWT/RE05/slides/09_Non-functional%20Requirements.pdf
[6] Ivica Crnkovic, Stig Larsson, Michel Chaudron," *Component-based Development Process and Component Lifecycle*",http://www.mrtc.mdh.se/publications/0953.pdf
[7] http://www.liacs.nl/~marcello/CBSE/development.pdf
[8] U.Banodha , K.saxena, "*Impact of Pipe and Filter Style on Medical Process Re-engineering*", International Journal of Engineering Sciences, October 2011.

[9] 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.
[10] 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.
[11] http://www.ncbi.nlm.nih.gov/pubmed/15306163
[12] U.Banodha, K.saxena," *Usability of Software Architecture Design Pattern in Medical Process Re-engineering Model*", International Journal of Application or Innovation in Engineering & Management (IJAIEM),Volume 2, Issue 6, June 2013, ISSN 2319 - 4847
[13] http://blog1000tutorial.blogspot.com/2009/07/chapter-1-business.html
[14] U.Banodha, K.saxena , "*A Software Architecture Style for Medical Process Re-engineering*" World Congress on Engineering and Computer Science 2011 Vol I WCECS 2011, October 19-21, 2011, San Francisco, USA
[15] U.Banodha, K.saxena, "*Dynamic Software Architecture for Medical Domain Using Pop Counts*", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 5, No.2, ISSN: 2156-5570(Online), 2014
[16] http://www.scialert.net/fulltext/?doi=jse.2010.16.29
[17] http://www.aljunic.com/aj/775/51
[18] http://blog1000tutorial.blogspot.in/2009/07/chapter-1-business.html
[19] http://www.csm.ornl.gov/~sheldon/cs531/ch8.pdf
[20] Lawrence chung, Kendra cooper, anna yi, "*developing adaptable software architectures using design patterns: an NFR approach*", computer standards & interfaces 25 (2003), 253-260, www.computerscienceweb.com.
[21] http://web.fsktm.um.edu.my/~azlina/wkes3202/NOTES_LEC/WKES3202_LEC8.ppt
[22] http://www.slideserve.com/poppy/se-565-software-system-requirements-non-functional-requirements
[23] http://bluehawk.monmouth.edu/~jwang/SE565_13.ppt
[24] http://www.es.mdh.se/pdf_publications/341.pdf
[25] http://bluehawk.monmouth.edu/~jwang/SE565_13.ppt
[26] http://www.iai.uni-bonn.de/III/lehre/vorlesungen/SWT/RE05/slides/09_Non-functional%20Requirements.pdf
[27] Lawrence Chung, Sam Supakkul, Anna Yi," *Good Software Architecting: Goals, Objects and Patterns*"

AUTHORS

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).

Diagrams

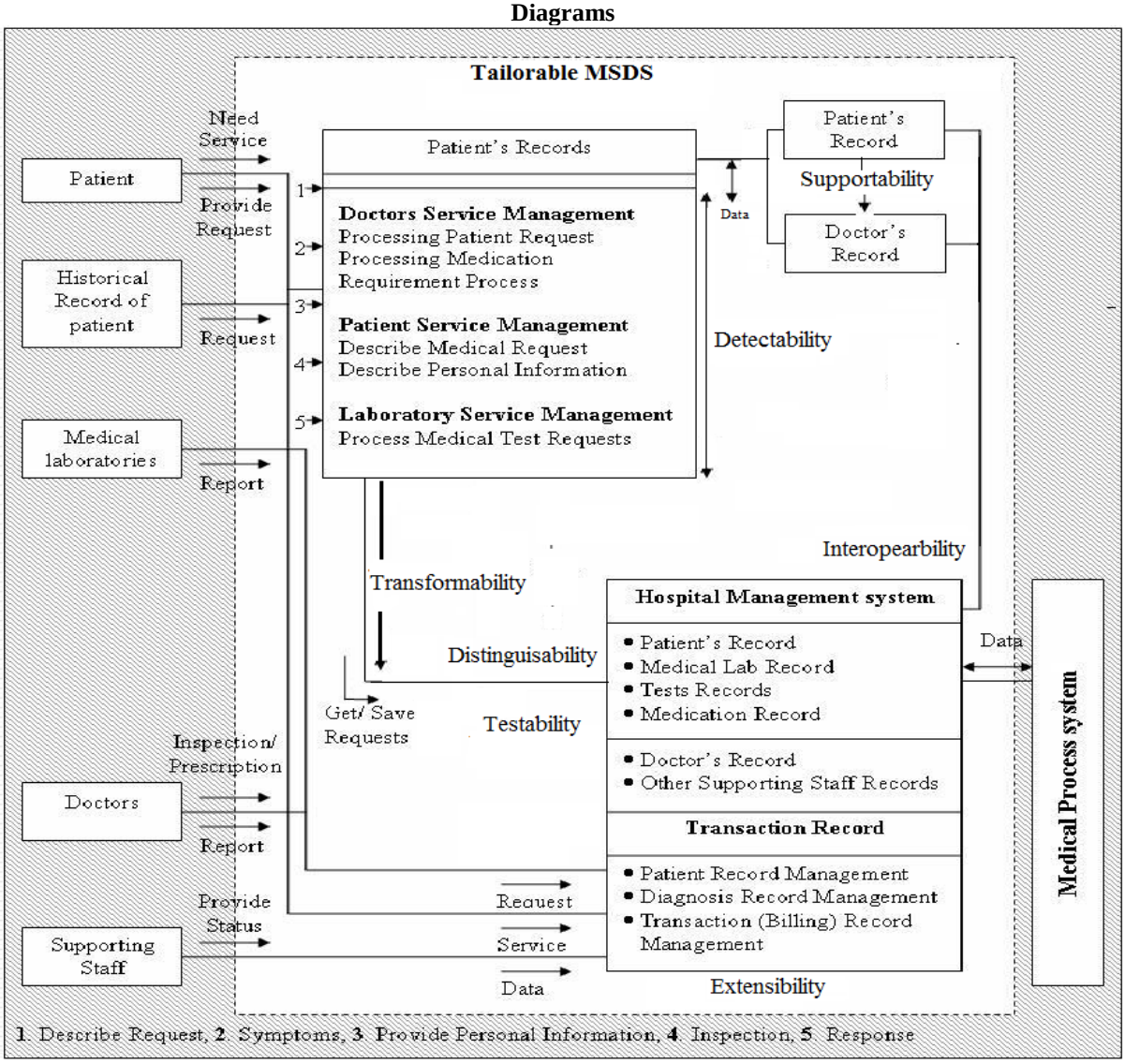

Figure 5 Tailorable Software Architecture using Non functional Requirements

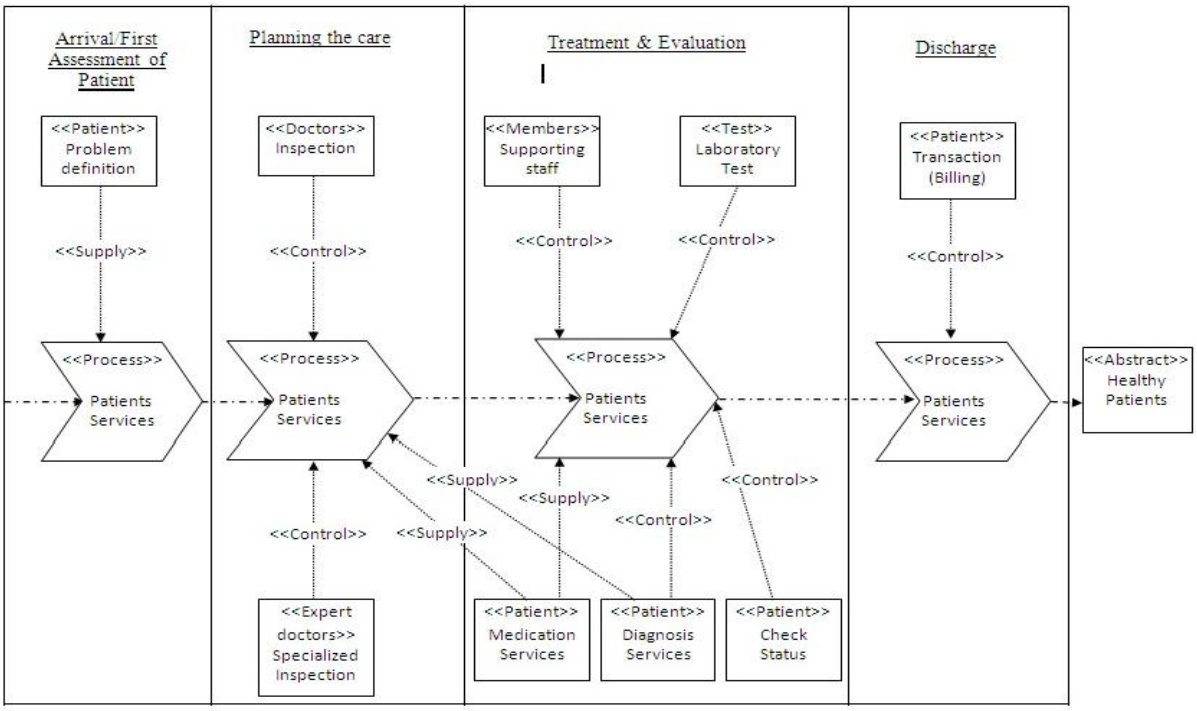

Figure 7 A Medical process diagram based on an activity diagram with Swimlanes

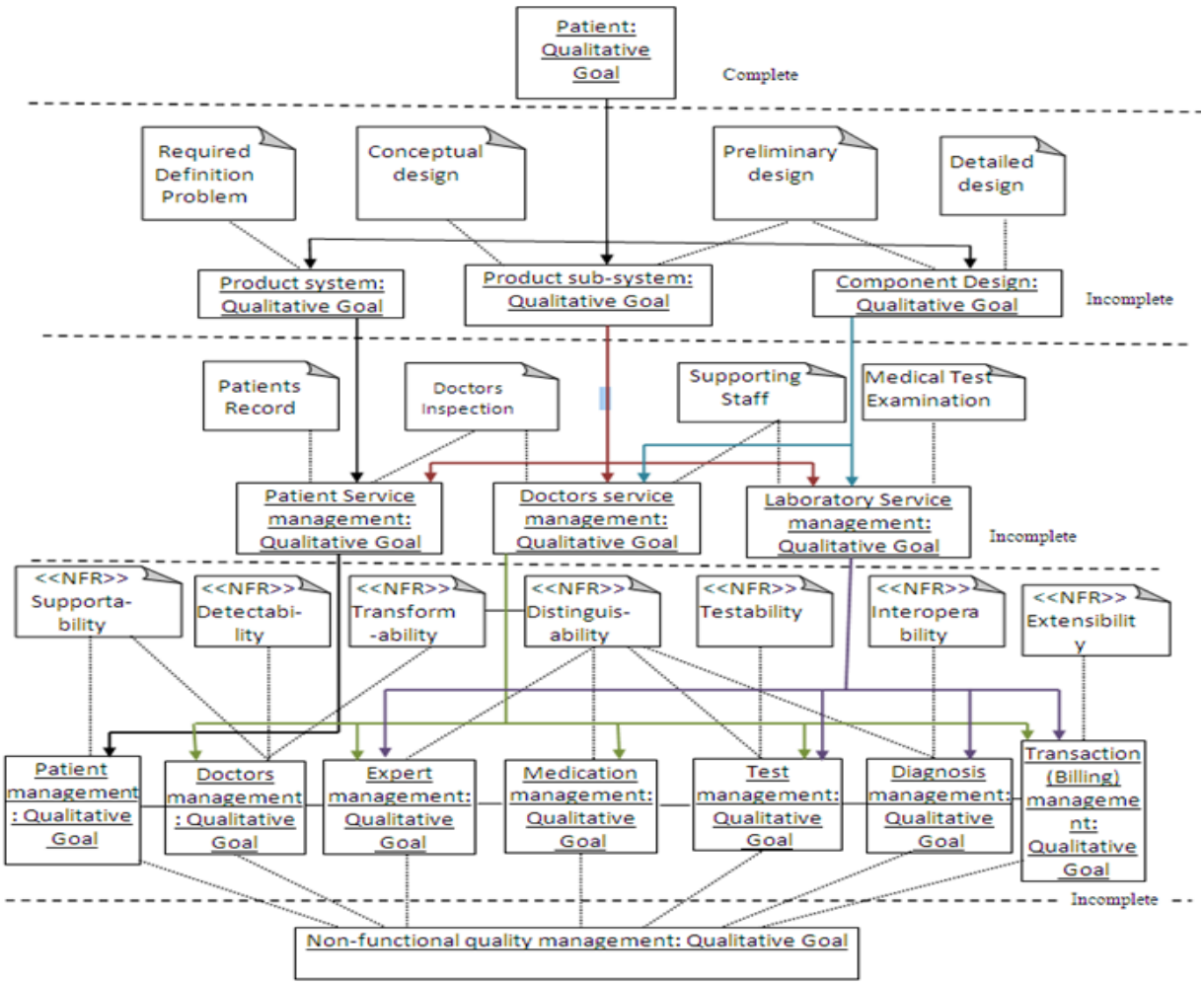

Figure 8 A goal-problem diagram based on a UML object diagram

