
Current Challenges and Opportunities of System Modeling for Hospital Automation

Author: Khaled SHUKRAN, Asia Europe Institute, University of Malaya, Kuala Lumpur, Malaysia, shukran_7@hotmail.com

This paper is focusing on the System Modeling for Hospital Automation, its challenges and opportunities. System modeling is mean to convert requirement analysis to system/specification. It shows the way of direction of how to use the system with effective way. It helps to understand system easier and builds the link among different activities and its impact. System modeling can trace out the problem area and after analyzing, it shows and determined the expected model, which could be more appropriate to implement for any organization, hospital as well as various types of business institutions. So the objective of this paper is to come across the system modeling for “Hospital Automation” and to find out the solution by analyzing with few methods. So this paper will discuss the domain scope for hospital, Software development life cycle, object oriented, function oriented and agent based software development methodology. It focuses also the requirement engineering. And finally also maps out the Jackson approach as well as VORD method for implement these approaches for Hospital Automation to meet the current challenges and accept the opportunities.

Keywords: System Modeling, Hospital Automation, Software Development Life Cycle, VORD method

Introduction

Hospital is a large social organization, which provides the health care service to the society. It deals with the human life what is a very sensitive and serious issue. Hospital is a complex system, which is not manageable in a simple method. It has Doctors and Nurses (the employees), Patients (customers), Board of directors (authority), Payment section and Payroll (finance division), security and many other divisions. Hospital provides the Medical treatments (service) to their patients through consultancies and prescriptions. In hospital, there are many categories and divisions in every sector. The Doctors are categorized according to their specialty such as Surgery, Medicine, Cardiology, Dental, Dermatology, Orthopedic, Gynecology, Nose-Ear-Neck and Neurology. The Nurses, administrative officers, authority and finance divisions are also categorized in different levels.

As it has many components or entities that are closely interrelated, dependent and associated with each other, it's really very difficult to maintain the whole process in a very simple way. If any component does not work properly then other dependent components can be hampered. For example, in a hospital if nurse-scheduling, causes the conflict or low skilled nurses are scheduled for serious patient then patients and doctors suffer many undesirable problems.

In present, manual paper based record keeping system of the patient information in the large volume of books are inconvenient. It's not convenient and takes long time to find out a particular patient records and medical history. In manual system everything is paper based that is very hard to maintain the total process. Even staff scheduling for both the wards and the operation theatre is difficult and its may creates conflicts in the manual system.

For the better management system of the hospital service, I believe, update of the computerized system is the best solution. The Database will contain all information about in-patients, outpatients, doctors, nurses, ambulance, Operation Theater, Ward/Cabin, medicine under a specific domain. In this system it's easy to add, change, or delete data from the database, correct information in time, accurate the billing system and easy payments for the better cash flow. Security is another important issue for

the hospital, which can also be, maintain by this system. Also, I would like to generate the reports of the total information about the doctors, nurses, the ward booking, the ambulance and the OT booking, admission of patients in a month, the financial condition of the hospital, particular patients bill, and other statistical reports. This information's are very important for the better hospital service.

So, I think, automation is absolutely needed for smooth functioning of the total process. It will be more time consuming and easily maintainable. To develop the total idea I am going to implement few models in this paper, which are effective for the hospital automation process.

Domain area or domain scope:

Now I would like to discuss about the domain area or scope, which I want to automate by the computerized system.

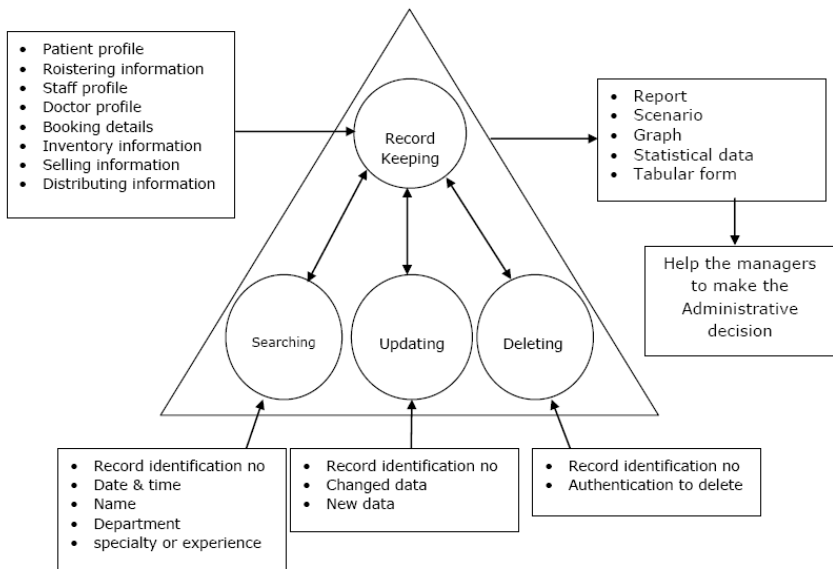


Figure 1: Our problem scope of a Hospital

The above Figure 1 illustrates the problem area of Hospital Automation. The input of the record keeping process of this manual system is patients profile, fostering information, staff profile, doctor profile, booking

details, inventory information, selling information, distributing information, scheduling information. This process is completed manually in books.

The patient profile contains the particulars of the patient like identification number, patient's name, age, sex, Father/Husband name, contact address, blood group, the patients category (in-door or out-door) etc.

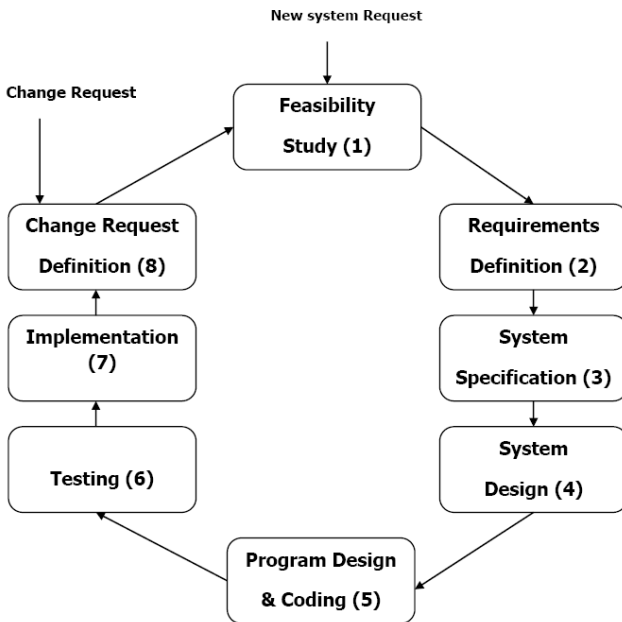
The contents of the doctor profile are the doctor recognition number, doctor's name, designation, contact address, contact number, department etc. The staff profiles are the combination of all information about a staff as like as a doctor profile. Here staffs include nurses, receptionist, matron, pathologist, pharmacist, medical officer. The inventory information are stored with inventory items id, name, price, quantity, and date of inventory i.e. purchase date, supplier. The inventory item includes the medicine, OT apparatus, scraps, Lab specific accessories, bed etc. The selling information is also having same data like inventory items like sold item id, item name, quantity, price, date of sale etc. In Fig-1, the searching process finds and shows the desired particular record by taking the record identification number or date & time. In the manual system it is too hard to show searching result successfully. Also, Fig-1 illustrates the updating process that can update a record by using identification number. Here the deleting process deletes the record from the database that is not required for further operation.

After completing these processes, some statistical data like number of patients admitted in a month, monthly income and expenditure, monthly medicine consumption, type of diseases and patients, number of empty wards and beds, monthly medicine and accessories stock etc. This paper will also represent this information by graph such as line graph, chart (pie chart or bar-chart) or by the graphical representation of data. These statistical data can help the managers, directors and executive directors to take proper administrative decision for quality service. The manual paper based system cannot provide these facilities so easily. Therefore, to develop the total system I have to take the system requirements, which is discussing in the next part.

Methodology:

There are different types of methodology has been used for system modeling. Now a day's organization are forming and adapting for client oriented. So different types of system development or software development life cycle have been built or created to manage the complexity of the organization. „The term life cycle indicates the staged nature of the process“ (Avison& Fitzgerald 2003). Therefore, SDLC is a sort of methodology which builds the process of information system with a good structure. In this part I am going to discuss about the SDLC life cycle and different types of methodology like *Object Oriented Software Development Methodology*, *Function- Oriented Software Development Methodology* and *User oriented Software development methodology* and *Agent based Software development Methodology* (Ayodele, Oluyomi 2007). It will focus why these types of method have been taken for and its application.

Software Development Life Cycle (SDLC):



1. **Feasibility study:** Feasibility study provides a clear statement of the purpose of a proposed system. It discusses what the problem is possible, how something might be achieved and how much effort will be required to solve the problem.
2. **Requirement definition:** Requirement Definition produces a detail and accurate description of the problem and the needs of the stakeholders of that system. It also ensures that all the requirements are identified and no important requirements are forgotten or unrecorded.
3. **System specification:** System specification integrates the processes, interfaces and data definitions that support all the stakeholders' requirements. It also concentrates what a system must do and how the system might be achieved.
4. **System design:**
5. **Program design & coding:**
6. **Testing:**
7. **Implementation:**
8. **Change request definition:**

There are many methodologies to construct these steps but there is no fixed methodology for each step. It may vary in respect of the problem domain.

Different Methodologies:

Object Oriented Software Development Methodology (OOSDM): OOSDM is widely used approach to problem solving takes an object oriented viewpoint. Here the problem domain is characterized as a set of objects that have specific attributes and behaviors and that are categorized into classes and subclasses.

A formal definition of object oriented, introduced by Coad and Yourdon [COA91], is:

$$\text{Object-Oriented} = \text{Objects} + \text{Classification} + \text{Inheritance} + \text{Communication}$$

The main features of OO Method are:

- **Information Hiding:** A well designed OO system always encourages information hiding. It hides the operational details of program component. Only the components that are allowed to access such information which are accessible only for those components.
- **Data Abstraction:** this is the mechanism that focus the essential focus the details of the program components(data or process)
- **Encapsulation:** dynamic binding: binding at runtime, polymorphism, virtual functions
- **Inheritance:** incremental changes (specialization), reusability

The object-oriented software life cycle

- Analysis -- Conceptual Model, System Requirements
- Design -- System Design, Detailed Design
- Implementation -- Coding, Testing

Function Oriented Software Development Methodology (FOSDM):

Function-oriented (or procedural) design decomposes the design into a set of interaction functions which act on a centralized state. This approach concentrates on the processing and algorithms of the system. Function-oriented design has been practiced since programming began, and there is a large body of design methods based on functional decomposition. However, while the functions hide the details of the algorithms, the shared state can be a particular problem as a function could change the state in ways not anticipated by other functions. It is commonly believed that object-oriented approaches (which also hide the state) are preferable. But function-oriented design is still important: In systems with minimal state (e.g. ATM), or which can be implemented by parameter passing, object-oriented approaches offer no significant advantages, and may even be less efficient. Many organizations have standards and methods based on functional decomposition. There are an enormous number of legacy systems out there, which have been developed using a functional approach, and need to be maintained (e.g. Windows, Linux).

User-Centered Software Development Methodology (UCSDM):

User-centered design has numerous benefits for business. *Firstly*, UCD methods result in higher-quality screen-based systems with increased customer satisfaction and confidence. In the competitive online market, a high-quality design can mean the difference between success and failure. It's a fact that customers abandon sites that are cumbersome and do not meet their needs. On the other hand, customers remain loyal to sites they can navigate. Forrester reports that 90% of customers would shop again at sites that met their expectations and 87% would tell friends and family about the site.

Secondly, under UCD methodologies, software development is more efficient. The system will likely go to market faster and cost less. The following table demonstrates that the average time required to fix a problem with software design or specifications is quite long and therefore very expensive. With a UCD approach, the product's design and specifications are frozen late in the development process. The method's constant customer feedback and design iterations weed out potential problems before they require hundreds of hours and thousands of dollars to fix.

Agent-Based Software Development Methodology (ABSDM):

An *agent* is an encapsulated computer system that is situated in some environment and it is Capable of being flexible. In order to meet its design Objectives autonomous action has been taken (Jennings, 2000). There are two fundamental concepts associated with any dynamic or reactive system, such as an agent, that is situated in and reacting with some environment (Holcombe & I pate, 1998):

An agent-based system is a complex software system with functional and nonfunctional constrain. Designing and building such system is a complex task. A goal-driven use case approach has been taken for agent-based system requirements analysis. The use case approach is used to elicit system requirements from user's point of view. Related use cases are assigned to corresponding roles. Each use case is then extended with goals for implicit requirements analysis from a role's point of view. Each role is treated as internal actor to find more system specific use cases. Five relationships between use cases and goals are then identified: satisfied, satisfiable, denied, deniable, and independent. Those relationships help find

the relationship among roles. Such relationships can be classified as cooperative, conflict, counterbalanced, and irrelevant. Identifying those relationships assists the system analyzer to analyze and optimized the relationships among roles. The contribution of this paper is a proposal to a systematic approach for implicit requirements analysis for agent-based systems.

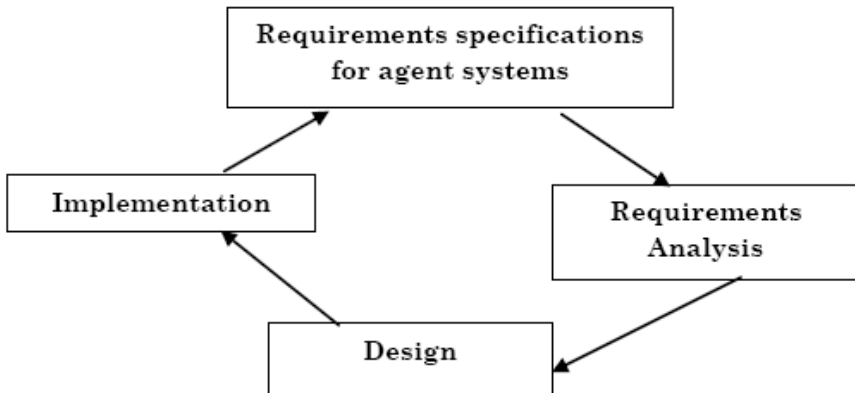
Agent-Oriented Software Engineering is an approach to developing software using agent-oriented abstractions:

- Agents
- High-level interactions
- Organizational relationships

Why it is need:

- Agent-oriented decomposition is an effective way of partitioning complex problems
- The key abstractions of agent-orientation is a natural way to model complex systems
- The agent-oriented approach to dealing with organizational relationships is appropriate for complex systems.

Agent-Oriented Software Life Cycle



Some General Application Areas:

Industrial applications

- manufacturing
- process control
- Air Traffic Control
- telecommunications
- transportation systems

Electronic Commerce

- electronic markets/auctions
- Buying agents (e.g. Jango, shopbot, etc)

Business Process Management

Information Management

- information gathering
- information filtering

Medical Applications

- Patient Monitoring
- Healthcare

E-Learning

- Intelligent Tutoring Systems

Requirement Engineering:

In this part, it will be focusing about the requirement engineering and its importance. Finally, it will focus its practical application for Hospital Automation.

Requirement Analysis:

Definition: the process which establishes the services the system should provide and the constraints under which it must operate is called Requirement Engineering (RE).

- Why Requirement Engineering is important?

- Importance: RE is first, major and time consuming part of SDLC. Analysis says 67% effort have to give for RE of whole SDLC to develop a **dependable** system and the rest 33% effort have to give for others steps – design, coding, testing and maintenance.
- If we capture the system, information by a well defined RE methods 67%work will complete. Error on the next step will be dramatically reduced. Consequently maintenances cost will be far less & system will be dependable.
- Another statistics say error for the requirement analysis (RA) is 56%,
- For design 27%, code 7%, other 10%.

So, well Requirement engineering (RE) is important to develop quality System.

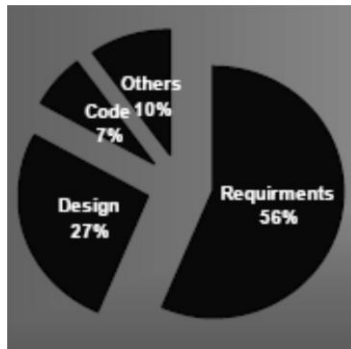
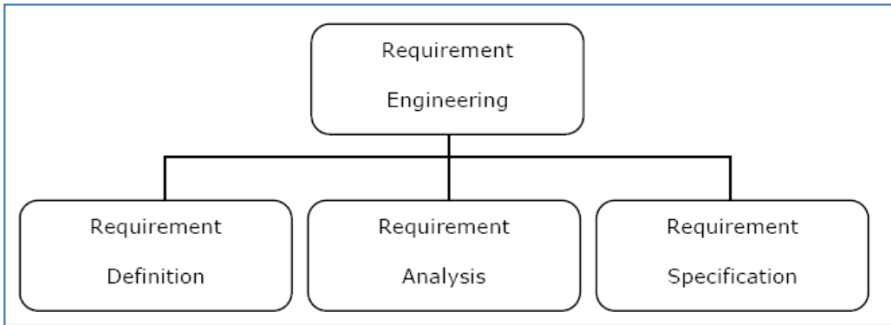


Figure 2: Errors in the development life cycle

Requirement Engineering (RE):

Requirement Engineering is an activity that transforms the needs and wishes of customers and potential users of computerized systems usually incomplete and expressed in informal terms into complete, precise, and consistent specifications is preferably written in formal notations.



Result (Application of the above four methods):

Now I would like to implement all of these methods in our problem domain:

Hospital Automation

Problem Domain Characteristics:

In this part, it will be discussing the characteristics of our problem domain to analyze and specify system requirements successfully. If the system requirements are not captured accurately, the proposed system will not run successfully.

The domain characteristics include- environmental characteristic, behavioral characteristics, and *system characteristics* [C. G. Davies].

Here, in this part it will be focusing the domain characteristics as follows:

Environmental characteristics:

In the environmental characteristics, it is concerned with recording general information, which characterize that domain- the interested users or stakeholders in the domain, the fundamental nature of the problem, and the general environmental factors and constraints.

The Figure 1 shows the main components of our problem domain. Now it will be examining in turn.

Stakeholder list:

Stakeholders and system components are the interested parties of the proposed system. In the Hospital Automation system example, the possible stakeholders would include- the receptionist, account officer, medical officer, matron, OT in-charge, pharmacist, pathologist, storekeeper.

The Table-3.1 shows the stake-holder list of Hospital Automation (HA) problem with their status (primary, secondary and tertiary), functions, objectives, association with others users and their problems. The status will give information about the type of the stakeholder.

For example, in the **column (2)**, the receptionist is the primary user that is s/he interacts with the system directly.

The objectives and functions of users assist the system developer to identify the system events and their information requirements very easily. For example, the functions [**column-(3)**] of the receptionist are patient’s registration and admission, patient query and discharging of indoor patients. And the objectives [**column-(4)**] of the receptionist are preserving previous records, maintaining released patients information, and collecting payments from patients. The association with the stakeholders and potential users [**column-(5)**] are recorded for identifying the relationship with the users that helps the analyst to create **Entity-Relationship** diagram. For example, the receptionist is dependent with account officer for billing purpose, pathologist for report collecting, matron for checking bed availability, and medical officer for patient discharging. And the problems [**column-(6)**] are identified for modeling the domain and providing system functions. For example, the receptionist faces the problems like missing previous record, maintaining admission serial number, getting wrong information of patients.

Table 3.1: A sample-recording schedule for information about Hospital Automation (HA) stake-holders / Users:

Stakeholder (1)	Status (2)	Function (i.e. jobs tasks) (3)	Objective (i.e. why?) (4)	Associate With other Users (5)	Problem (6)
Receptionist	Primary	<ul style="list-style-type: none"> - Registration - Admission -Report collection & delivery - Patient query - Discharge of in-patient 	<ul style="list-style-type: none"> - To preserve previous Record - To get better word / bed - Get Patient information - Released patient info. - Collect payment 	<ul style="list-style-type: none"> - Account officer - Pathologist - Matron - Medical officer 	<ul style="list-style-type: none"> - Missing previous record. - Admission Serial maintain - Patient fail to payment bill - Get wrong information from illiterate patient.

Environmental Factors and Constraints:

The Table-3.2 illustrates the environmental factors and constraints’ references, their description and impacts on organization to describe the constraints and non-functional requirements of the domain. It also describes the impacts on organization. This kind of information is so important because it will affect a variety of design decision to be made in the development process. For example, one of the reference environmental constraints is drug law. This is described in **column-(2)** and how it affects the hospital in **column-(3)**.

Table 3.2: Environmental Factors and Constraints occurred in Hospital Automation (HA)

Reference (1)	Description (2)	Impacts ... (3)
Drug law/ BSTI	BSTI marks some medicines that are prohibited for marketing and consumption.	This medicine cannot store and medicate in the hospital. Hospital authority will be responsible for bad impact of these medicines.

Problem / Requirement List:

The **Table-3.3** shows the problem name, problem symptoms, impact on organization, cause of the problems and problem priority. The problems are those that are identified in the stakeholder list. The symptoms (who or what are effected) [**column-(2)**] are discovered to identify the root of the problem to be solved. For example, the symptom of the problem- missing previous record is occurred when patient needs previous record when he/she comes more than once. The form also describes the impact [**column-(3)**] on organization that helps us to determine what problems are more or less harmful for the organization. For example, if the receptionist faces the problem- missing previous record occurred then the data may redundant. Next, it will try to find out the cause of the problem to resolve and finally, prioritize the problem so that problems with highest priority have to be solved immediately. For example, for those problems the register book may be very large that is difficult to maintain.

Therefore, the overall objective of this table is to provide a scope for the modeling of the real world problem and identify the user information generating functions.

Table 3.3: Problems / requirements list of the users

Name (1)	Symptoms (who or what are effected) (2)	Impact on organization (3)	Cause of Problems (4)	Priority (5)
<ul style="list-style-type: none"> - Missing previous record. - Admission Serial maintain - Patient fail to payment bill 	<ul style="list-style-type: none"> -When patient need to come more than once, they need previous record. -VIP or some people try to take early appointment. -Request for concession 	<ul style="list-style-type: none"> -Redundant information so high maintenance cost -Maintenance problem -Loose profit 	Register book is large or may have many volume that is difficult and time consuming matter	1

Behavioral Characteristics:

After describing the environmental characteristics, it will be discussing about the behavioral characteristics of the system. It includes *object inventory*, *process inventory* and *event list*.

Object Inventory:

The **Table 3.4** shows object inventory of the problem area which includes name, description, dependencies or associations with other objects, access, availability and location, possible states, and processed by of objects.

For each object, a brief description is given with their dependencies or associations with other objects that helps the analyst to construct graphical representation of the associations between objects such as: **Entity-Relationship model** etc. Access, availability and location describe the accessibility and availability of the objects. Finally, the possible states and process information help the analyst to identify and describe the states and processes of the system that will eventually cross-reference with the event list and process inventory respectively.

For example, the object patient has a relationship with doctor, nurse, and its possible states are enquiry, change, and discharge. And states are processed by patient registration, admission and discharging.

Table 3.4: Identified Objects Inventory of Hospital Automation (HA)

Object name (1)	Patient
Description (2)	Who is admitted or registered in the hospital for long or short term treatment.
Dependencies/Associations with other Objects (3)	Doctor Nurse Receptionist Account officer
Access, availability and Location (4)	Patient information is maintained in a master book record. Each patient has a reg. no as shown in the appropriate patient record book
Possible States (5)	Enquiry Change Discharge
Processed by (6)	PatientRegistration Patient Admission PatientDischarging

Process:

The **Table 3.5** shows the processes that change the states of the objects of Hospital Automation (HA) and their description, users, related objects and start/stop events.

Firstly, this form contains the process name that comes from Object Inventory table and this process is the part of the function of a department. Next it shows the process and associated users and objects that determine who starts the process and where. Finally, it starts and stop events are also recorded for process triggering. Therefore, the aim of these information is to begin understand the dynamic and behavioral aspects of the domain.

Table 3.5: Processes of that can change the states of objects for Hospital Automation (HA)

Process name (1)	Part of function... (2)	Description (3)	Users (4)	Objects (5)	Start/Stop Event (6)
Patient registration	Receptionist Desk	Fill-up the patient's information form.	Receptionist	Patient registration form	Start: entry patient's information. End:

Event List:

The **Table 3.6** shows the events name and their description, caused by an association of the process. After listing the name of the event, this form describes the event and cause of the event i.e. when and why the event is triggered. The final aspect of the problem domain's activity description to be recorded is the events, which occur in the domain. The rule for identifying events is that they must be at an instantaneous point in time and result in the changes of an object.

Table 3.6: Event list that trigger the processes of Hospital Automation

Event name	Description	Event caused by	Associated functions/Processes
Patient register	The point of time at which a person can be a registered patient of that hospital.	Patient information entry for treatment	Patient registration

System Characteristics:

After describing behavioral characteristics, it will discuss about the system characteristics. It is important to note that it is only the characteristics of the solution, which are being considered. At this stage, it is tried try to identify the general characteristics of a solution.

System characteristics contain *information generation requirements, system function outline* and *system transaction*.

Information generation requirements:

The aim in considering the information generation requirements of a system is to attempt to define the overall purpose and goal of proposed system.

The **Table 3.7** shows the name of the output report, together with a description, the process that is associated and the user for whom the information is intended and the contents of the report. For example, the report named patient registration details is generated by patient’s registration process and this process is triggered by receptionist. This **Table 3.7** shows the contents of that report.

Table 3.7: Information Generation Requirements recording schedule

Report Name	Generated by process	For user.	Contents
Patient Registration Details	Patients registration	Receptionist	This report contains all information about patients such as ID, Name, age, sex, Father/Husband name, Address, References.

System Function outline:

The **Table 3.8** illustrates the sample system function outline. For each proposed system function outline the main objective of the function are listed, together with an outline of processing the developer can start to identify the particular transaction to be supported by system. For example, this table shows the registering function with its objective- preserving all information of patient, and outline processing- preservation of information requires facilities for medical history and maintaining patient service. The problem of the function described in the **column-(4)** is awareness of previous & present record. This table also explains the problem of the function and the users who execute the function. For example, the registering process is triggered by front desk officer- receptionist.

Table 3.8: system function outline recording schedule

Function Reference (1)	Objectives (2)	Outline/ Processing (3)	Problem addressed (4)	Users (5)
Registering	Preserving all information of patient to provide service.	-Preservation of information requires facilities for medical history -maintaining patients service	Awareness of previous & present record	Front office desk (Receptionist)

System transaction:

The **Table 3.9** shows the sample of system transaction for Hospital, which describes the users, transaction name, precondition and description. These transactions summarize the function to be included in the network phase of the system development and cover both input and validation.

For example the transaction registration is initiated by receptionist and its precondition is recommendation of medical officer and the description of the transaction.

Table 3.9: system transaction for Hospital Automation (HA)

User	Transaction name	Precondition	Description
Receptionist	Registration	-Recommendation of Medical Officer -Availability of respective doctor -Availability of ward/bed -check sponsor status	The system response with these preconditions for patient registration. If the criteria are not match, the proposed registration cannot be performed.

OMT Method:

OMT (Object Modeling Technique) is an Object Oriented that consists of three kinds of models- *i. Object Model*, *ii. Dynamic Model* and *iii. Functional Model*. It is applied all of these models in our problem domain below.

Object modeling:

In the first step, object model analyzes the requirements to describe real world object classes and their relationships to each other. To construct an object model the following steps are performed:

- **Identify objects and classes:**

The first step to construct an object model is identifying relevant object, classes from the problem statement. All classes must make sense in the problem domain. In our problem domain, some of the classes are identified from the problem statement in the following:

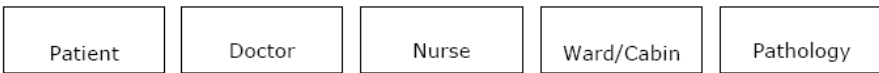


Figure 5.1: A few objects and classes of Hospital

The above figure named Figure 5.1 shows some classes and objects- patient, doctor, nurse, ward/cabin, pathology. Patient who is registered or admitted to hospital for treatment, Doctor who serves the patients, Nurse who is helping hand of doctor, ward/cabin where patients live during treatment and Pathology in which patients test their body’s specimen to determine the diseases they are suffering.

- **Prepare data dictionary:**

Data dictionary is the precise description of all modeling entities. It describes the associations, attributes and operations. Here is a data dictionary of above objects.

<p>Patient: Who takes the services from the hospital?</p> <p>Doctor: the employee or visitors who investigate the patient’s diseases and give the proper treatments for the patient.</p> <p>Nurse: the employee who is helping hand to the doctors and nursing to the patients.</p> <p>Ward/Cabin: that consists of several beds where the patients stay and take the services from hospital. Ward/cabin can be deluxe, super deluxe, economy.</p> <p>Pathology: the department where all diagnosis tests are performed and create report.</p>

Figure 5.2: Data dictionary of some classes of Hospital Automation

- **Identifying the right association:**

Next, is to identify the right associations between classes. Any dependency between two or more classes is an association. For example, in the [Fig- 5.2] the association within patients and doctor is patient need ward/cabin to staying in the hospital

- Patient needs ward/cabin to staying in the hospital
- Patient gets test report from the pathology department
- Doctors gives treatment to the patients
- Nurse helps the doctors.
- Pathology tests the all kinds of diagnosis tests.
- Patients get the bill when s/he discharged.

Figure 5.3: A few of right associations of Hospital Automation (HA)

- **Identifying Attributes:**

Next, the Table 5.1 shows the identifying objects with attributes. For example, patient is a object with attributes Patient ID, name, address, age, sex, father’s/husband name, blood group etc.

Table 5.1: The attributes of the objects identified from the Hospital Automation (HA)

Objects	Attributes
Patient	ID, name, address, age, sex, father’s/husband name, blood group
Ward/cabin	Ward ID, Ward Name, bed ID, types of ward/cabin, rent of bed.
Doctor	Doctor ID, name, designation, visiting time, contact #, address
Nurse	Nurse ID, name, category, working session, contact #, address

The Figure 5.4 shows the Object model of Hospital with some of the objects and their attributes. In this figure patient is depend on doctor for getting treatment and doctor is depend on nurse for taking help and patient is depend on nurse to get better nursing.

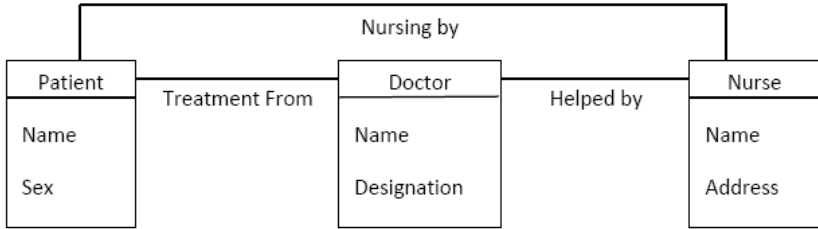


Figure 5.4: Hospital object model with attributes

- **Refining with Inheritance:**

The next step is to organize classes by using inheritance to share common structure. Inheritance can be added in two directions by generalizing common aspects of existing classes into a super class (bottom up) or by refining existing classes into specialized sub classes (top down). It can be discovered inheritance from the bottom up by searching for classes with similar attributes, associations, or operations for each generalization; define a super class to share common features. **For example**, in the [Figure 5.5] ward booking, OT booking and ambulance booking are similar except in their initiation and can be generalized by booking.

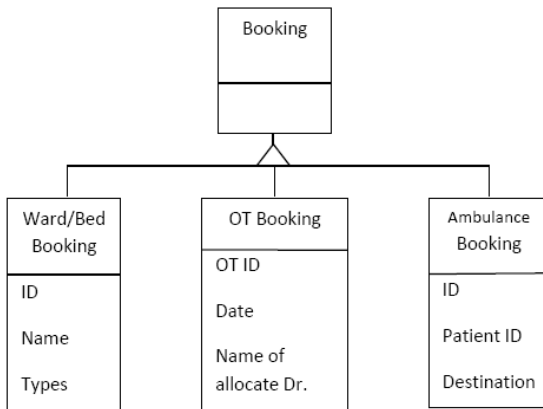


Figure 5.5: Hospital object model with attributes and inheritance

Dynamic:

The dynamic model shows the time dependent behavior of the system and the objects in it. From the Beginning, dynamic analysis is looking for events. Then summarize permissible events sequences for each object with a state diagram.

- **Preparing a scenario:**

Preparing one or more typical dialogues between user and system is to get a feel for expected system behavior. This scenarios show the major interaction, external display format and information exchange. A scenario is a sequence of events. An event occurs when an information exchange between an object and outside agent. For example, the [Figure 5.6] shows a scenario of major interaction within the objects patients and receptionist. Here receptionist asks the patient's information to register or admit.

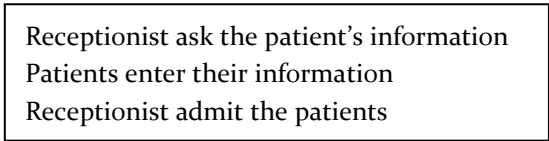


Figure 5.6: normal Hospital Automation (HA) scenario

- **Identifying events:**

Examine the scenarios to identify all external events. Events include all signals, inputs, decisions, interrupts, and actions to or from users or external devices. For example, the [Figure 5.7] illustrates one event- Receptionist registers patient that gives the knowledge of actions, decisions and objects of the event.



Figure 5.7: One event of reception desk of Hospital

Event trace diagram:

Event trace is an ordered list of events between different objects assign to column in a table. The [Figure 5.8] shows an event trace for a patient registration scenario. If more than one object of the same class participates in the scenario assign a separate column of each object by scanning a particular in the trace, it can be seen the events that directly affect a particular object. Only these events can appear in the state diagram for the object.

For example, when the patient enters to hospital, receptionist requests that patient’s profile and enters that profile. If name and date of birth are matched then return previous ID otherwise create new ID and stores the records and registers the patient successfully.

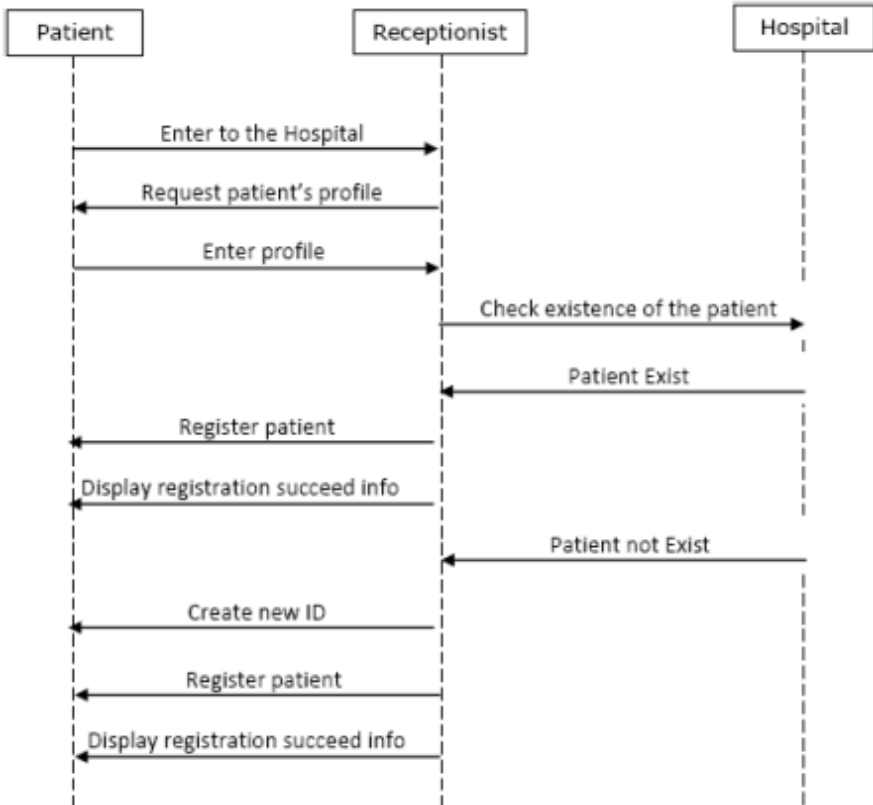


Figure 5.8: Event trace diagram of reception desk of HA

Event flow diagram of reception of Hospital:

The Figure 5.9 shows the events between groups of classes on an event flow diagram. This diagram summarizes events between classes, without regard for sequence. Include events for all scenarios. The event flow diagram is a dynamic counterpart to an object diagram. A path in the object diagram shows possible information flow; a path in the event flow diagram shows possible control flow.

For example, receptionist requests patient for profile and enter the profile to system. System checks name and date of birth. If they are matched with previous record then return previous ID otherwise create new ID. Finally, after registration it displays successful registration information.

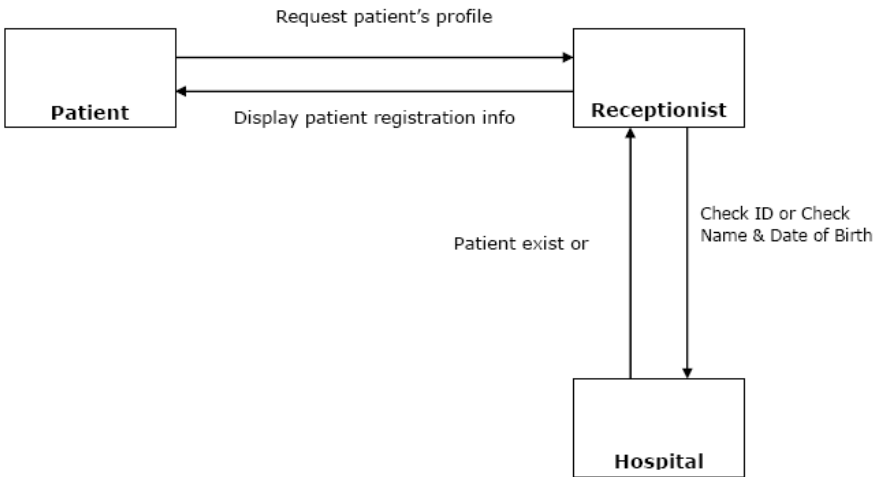


Figure 5.9: Event Flow Diagram of an event

- **State diagram of receptionist of HA:**

Preparing a state diagram for each object class with nontrivial dynamic behavior, showing the events the object receives and sends. Every scenario or event trace corresponds to a path through the state diagram. For example, the [Figure 5.10] illustrates the state diagram of registration process. Here when registration request is made then enter registration information and verify name and date of birth. If patient exists then return previous ID otherwise create new ID and then complete the registration.

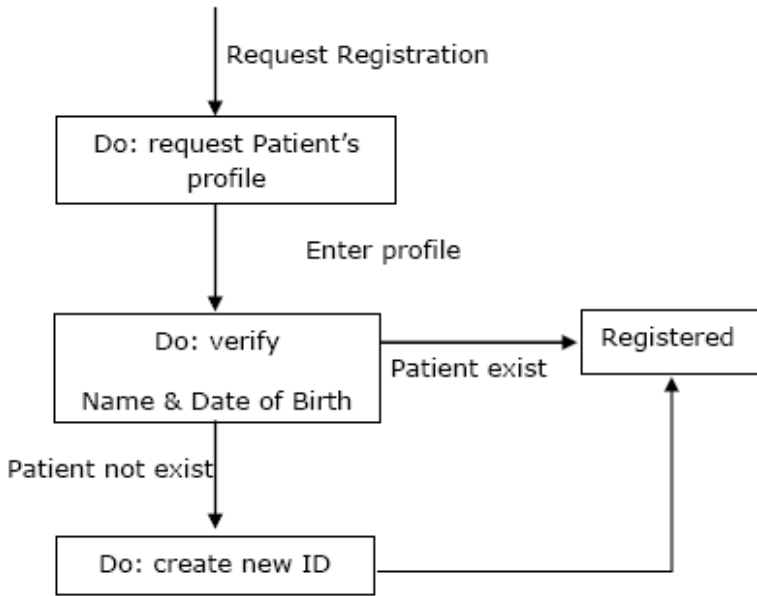


Figure 5.10: State Diagram of Receptionist of Hospital

VORD method:

The VORD method [Somerville, 1992] principally detects and analyzes the requirements and translates into Object Oriented system models by viewpoint identification, viewpoint structuring, viewpoint documentation and system mapping. This paper is tried to analyze the problem domain, a portion of the Hospital, and translates them into system models by following the VORD Method mentioned below.

Viewpoint Identification:

The stage in the viewpoint analysis of the VORD method is the identification of possible viewpoints/users and their services. This stage is probably difficult in all other methods. The brainstorming is an approach to identify the viewpoints and services that interact with the system. The following [Figure 5.11] illustrates the viewpoints and their services where the **white** ovals are viewpoints and the **shadow** ovals are services.

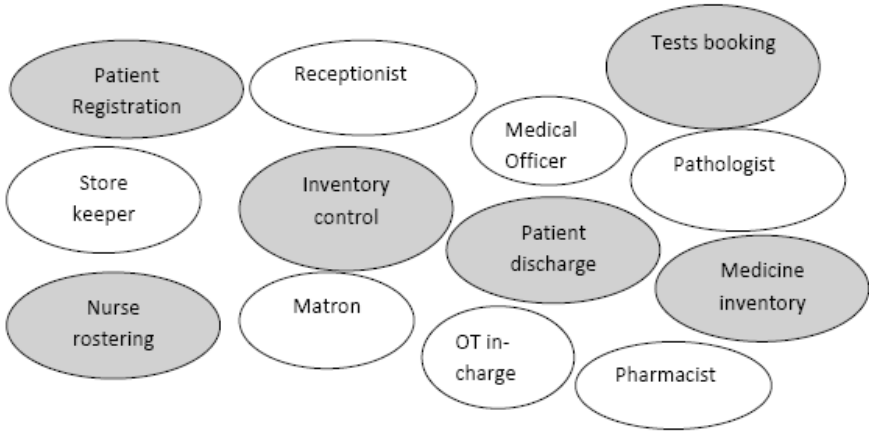


Figure 5.11: Brainstorming for Viewpoint Identification for Hospital

Viewpoint structuring involves grouping related viewpoints into a hierarchy. Common services are provided at higher levels in the hierarchy and are inherited by lower level viewpoints.

The [Figure 5.12] is the demonstration of the structure of the viewpoints that describes the services according to the viewpoints. Here the services may pick up successfully that helps the developers or analyst to identify the events, functions and processes with objects. For example, the Fig- shows matron is higher-level viewpoint of its lower level viewpoints Nurse in-charge and OT in-charge. Services of matron are recording doctor’s visiting time, keep medicine consumption record, patient transferred to desired ward check ward or bed availability. These are the common services of the nurse in-charge and OT in-charge. But their individual services are nurse roster, ensuring better nursing and maintain OT as per schedule, sterilization of all OT equipments respectively.

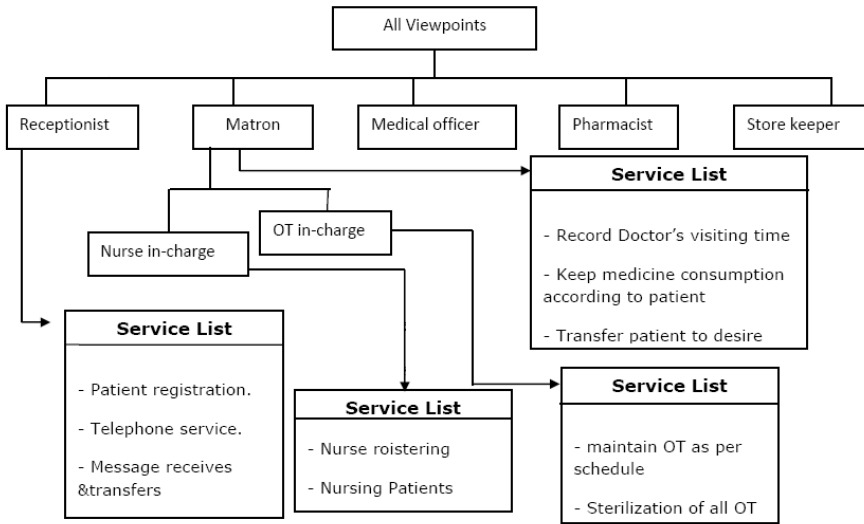


Figure 5.12: Viewpoint Structuring of Hospital Automation

Viewpoint documentation:

Viewpoint documentation concerns with refining the description of identified viewpoint and services. It includes viewpoint template and service template. It also includes the event scenarios.

Viewpoint template describes the reference, attribute, events, service, and sub-viewpoints of the viewpoint. For example, in the [Figure 5.13] the viewpoint reference, receptionist has attributes-EmpID, name, Sex, and working period. S/he performs the **start**, **end** and **cancel**s events of patient registration and discharge. The services are outpatient registration, in-patient admission and patient discharge and the sub-viewpoints of receptionist.

The service template of viewpoint explains the details of the services that include reference, rational, specification, viewpoint, non-functional requirements, and provider.

For example, the [Figure 5.13] shows the rational of the patient registration service is- to identify & preserve the previous patients & medical history or records. The specification of patient registration is- receptionist collects patients profile and then inputs to the system and receives confirmation.

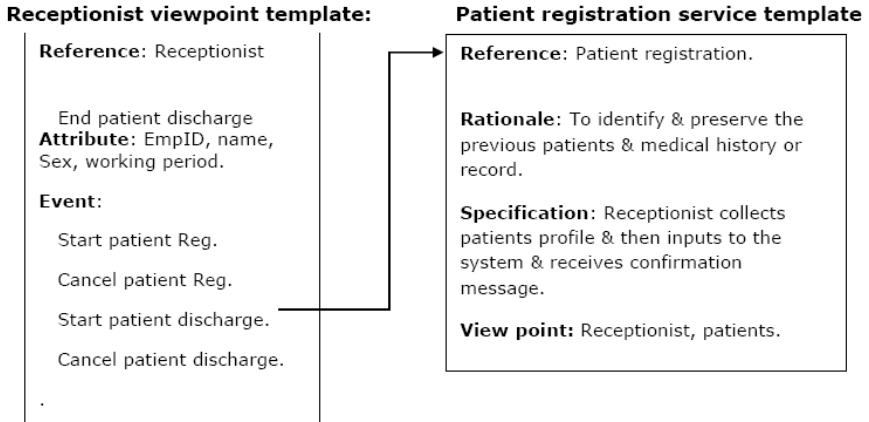


Figure 5.13: viewpoint template and viewpoint service template

The [Figure 5.14] demonstrates a scenario of an event, patient registration that realizes the total operation with inputs, processes and output. Here input is patient’s profile. The system processes the profile and takes only name and date of birth to check the existence of patient’s profile. If exist then returns previous ID otherwise create new ID and registers the patient.

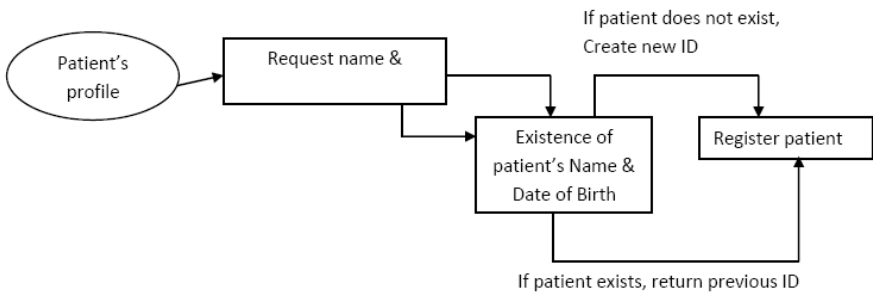


Figure 5.14: Scenario of patient registration event

Viewpoint system mapping:

The viewpoint system mapping specifies the requirements of system and shows the flow of data with process, entity, and file. In the [Figure 5.15] explains the flow of data among entities-patient, receptionist and medical

officer, process-registering patient and file-patient profile. It is a part of the system mapping.

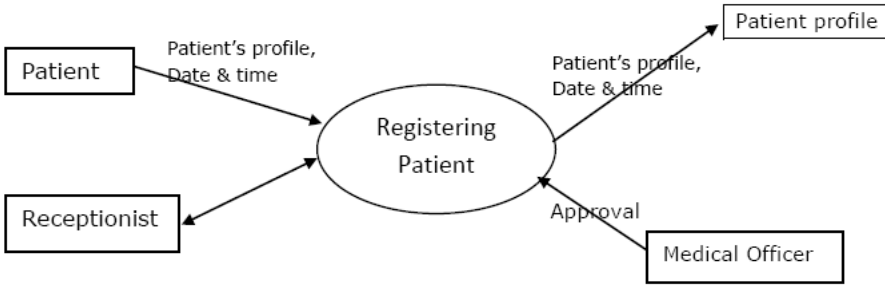


Figure 5.15: Data Flow Diagram of registering patient process

Discussion

On the basis of above results it can be observe that the four methods are functioned differently as their own way. But all of them do not support all stages of RE framework strongly.

In OMT method, it can be easily identified the possible objects. Identifying the right associations, attributes of the objects, creating object model with attributes and inheritance we can refine the objects. But making objects inheritance is bit difficult. So refinement of objects is ambiguous. But dynamic modeling is easier by preparing scenario, identify events, creating event trace diagram, event flow diagram and creating state diagram.

Jackson approach, analysis of the Requirements in various angles. First it identifies all the users and their functions, associations, problems. It measures the user status primary, secondary, tertiary. But other methods do not support this. After identifying all the problems, it analyzes their symptoms, impact on organization and their causes. It identifies the problem priority, but others methods do not prioritize the problems. After that it tries to identify the objects. But there are no obvious techniques to refine and extract exact objects. After identify the objects, it specify the associated process and events regarding the object all above are the part of Requirement Analysis (RA).

It performs, the Requirement Specification (RS) by the tables -- **Information Generation Requirements**, which identify the objects, attributes and related process, **System Function Outline** which broadly categorize the system function that user can perform with the system and hence provide a means for scoping the overall activities of the system.

It validate input, process by System Transaction and update the real world model, this transaction may be part of validation of input of system specification that corresponds the network stage of Jackson approach.

VORD method principally intended for requirement discovery and analysis. It also includes steps to help translate this analysis into an Object Oriented System Model. Information collected from different viewpoints is systematically transformed to object oriented design. The first stage of VORD Method concerns with viewpoints and services identification, viewpoint structuring and documentation.

Here viewpoints and services are identified by mainly brainstorming. Unallocated services can suggest viewpoints that have not been identified in the initial brainstorming session. It does not consider problem arises during the services provided the stakeholder. After viewpoints identification it performs viewpoints structuring with viewpoint hierarchy. It also prioritizes the requirements. But it doesn't set the priority problems (service problem) unlike Jackson approach.

OMT method also doesn't prioritize the service problem. It does not explicitly categorize the viewpoints. It shows the association of viewpoints by viewpoint structuring with their common and specific service. Jackson approach shows the association of stakeholder but how they interact / associate, it does not explicitly mention.

Third stage, viewpoint documentation is performed by viewpoint templates and service templates, which identify attributes, events, services of viewpoints and identify the causes, specification, non -functional requirements for each service.

During the first stage/Analysis phase, VORD method does not explicitly identified the objects. But in the fourth stage of system mapping they transfer viewpoint documentation into object model for Requirement Specification.

VORD method does not consider the validation of Requirement Specification (RS). In Requirement Analysis (RA) they don't discuss about process. Only in DFD, System Specification, It discusses about the processes before system mapping, system analyst ignores about the process.

Conclusions

From above review, it may observe that Jackson approach better follow the Requirement Engineering (RE) framework but object identification and their refinements are not perform in this method. Though object identification and their refinement for extracting the exact object is inevitable in Object Oriented method. It can be seen that OMT method provide better way for this approach, so our proposal is to incorporate Jackson and OMT for aggregating their better feature and excluding insufficient feature. As OMT provide better way of object identification and their refinement but Jackson provides all other Requirement Engineering (RE) framework stages. So it's may include this OMT feature to Jackson approach which may be better method for Requirement Engineering.

If we think only based on Requirement Engineering (RE) then Jackson approach may be better. If we think fully OORE, then OMT will be better. Though, we see that VORD method analyze the problem without regarding function or Object Oriented method. The out of analysis of VORD method can be translated in to Object Oriented Requirement specification named as system specification. In Jackson Approach class and inheritance is not defined but it discusses about association of how objects are associated.

References

- [1] Avison, David & Guy Fitzgerald(2003), *"Information Systems Development: Methodologies, Techniques and Tools"*, United Kingdom, Mcgraw Hills Publisher
- [2] I pate, Florentin & Mike Holcombe (1998), *"The Correct System"*, United States, Springer Verlage Publisher
- [3] Somerville, Ian (2000), *"Software engineering"* 6th ed., Pdf.
- [4] Chiung-Hui Leon Lee, Alan Liu, *"A Method for Agent-Based System Requirements Analysis"*, mse, pp.214, IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02), 2002
- [5] Coad, Peter & Yourdon Edward (1990), *Object Oriented Analysis* (2nd Edition), Prentice Hall, pp. 233
- [6] Oluyomi, Avodele, Karunasekera, Shanika & Sterling, Leon (2007), *"A Comprehensive View of Agent-Oriented Software"*, Autonomous Agents and Multi – Agent System, Volume 15, Number 3, pp. 337-377

-
- [7] Laura Bocchi, *"On the Impact of AOSE in Service-Oriented Computing"*, Emerging Methods Technologies and Process Management in Software Engineering, 02/02/2008

Supplementary Reading Materials:

- [1] Awad, Elias M.(1995), *"System Analysis and design,2nd ed."*, McGraw-Hill Publisher
- [2] Bruegge, Bernd & Allen H.Dutoit (1999), *"Object-oriented Software engineering"*, Conquering complex and changing systems
- [3] Colbert, Edward, (1993), *"Choosing the right Object Oriented Method"*, Pdf.
- [4] Davies C.G., & P Layzell, (1993), *"The Jackson Approach to System Development: an Introduction"*, United Kingdom, UK, Chartwell Bratt Publisher
- [5] Flynn, Donal J, *"Information Systems Requirements determination and analysis"*, 2nd ed. (2003). Mcgraw-Hill Publisher.
- [6] Rumbaugh, James R, Michael R.Blaha,William Lorensen,Frederick Eddy & William Premerlani (1990), *"Object Oriented Modeling and Design"*, Prentice Hall
- [7] Steve Skidmore (1997), *"Introducing systems Analysis"*, 2nd ed., Palgrave Macmillan Publisher