A Method to Specify Requirements for A Hospital Management Software

Mohammad Shahadat Hossain
Department of Computer Science, University of Chittagong, Bangladesh
Md. Abdul Halim
Md. Masudur Rahman
And
Khorsheed Alam
International Islamic University of Chittagong, Bangladesh

Abstract: The diagnosis of disease and the treatment, both medical and surgical, to the sick and injured people are the main role of a hospital as an institution. To achieve these objectives hospital is built in a suitable location, staffed with skilled manpower and equipped with the state of the art modern medical technologies. Nowadays, hospital as an institution is becoming an example of complex system because of the existence of increasing diverse interrelationships among its various entities/components. The management of the hospital, which is mainly manually operated, is unable to cope with this increasing diversification of the interrelationships and the information resulting form them. A computer-based information system (CBIS), which has the ability to handle large amount of information and their interrelationships in an organisation can be seen as an appropriate solution at this point. The development of an effective CBIS is not a straightforward task. This consists of a series interrelated stages such as requirement specification, system design, implementation and maintenance. Among them requirement specification is considered as the most important stage, since the viability of a CBIS mainly depends on it. Therefore, the paper presents the procedures of capturing requirement specifications for the CBIS to be used in a hospital. This can be demonstrated that the specifications in turn will enable an understanding of all aspects of the system in a consistent and coherent way and therefore to define its boundary.

Keywords: CBIS, System, Requirement Specifications, Hospital and Problem Domain
1. Introduction

The main objectives of the hospital is to diagnosis diseases and provide treatment to sick and injured people with trained manpower and modern equipments with an efficient and cost effective management. Diagnosis enables the detection of disease by diagnosis method disease are detected for proper treatment. It is carried out by collecting sample of body specimen, followed by its scientific analysis in a laboratory. The result of the analysis represented as a diagnosis report for the physician concerned with the patient treatment. For that case required body specimen is collected by giving test code according to test type and analyse them with scientific way than it provide the analysis result that is shown as diagnosis report provided to the doctor for appropriate treatment.

There are two types of treatment one is medical and the other is surgical. Treatment also be two types medical and surgical for Patient with normal disease may be subjected to the both types of treatment. In case of surgical treatment the patient has to stay at the hospital for a shorter period of time. Patient with serious disease normally undergoes for intense medical and/or surgical treatments. For both the cases the patient has to stay in the hospital for a longer period of time. The hospital provides housing facility to its patients, consisting of allocation of rooms, known as wards, and beds. Both types of treatments normal patient need not stay hospital but serious patient may have to stay hospital for long time treatments for that case hospital have to arrange housing facility that is built with words and beds.

Well-trained manpower and well-equipped instruments are necessary to perform To perform diagnosis and treatment activities. the job a number of equipments and manpower are ornamented who's efficient and effective interaction is necessary. Examples of manpower associated with the diagnostic activity are pathologist and sample collectors. Physician, surgeon, matron, nurse and midwife are the examples of manpower related to the treatment activity. The mentioned manpower are directly linked with the two types of activities diagnosis and treatment.

However, there are other examples of manpower, which are not directly related to the above activities but their role in functioning the hospital is vital. For example, the overall management of a hospital is dependent on a director. As a further example, account officer and his associated staffs are responsible for smooth functioning of the financial activity of the hospital. In addition, guards, and sweepers are responsible for maintaining security and a hygienic environment. The mentioned manpower can be linked with the activity management. Therefore, from the above it can be seen that in order to perform diagnosis and treatment activities, a hospital requires various types of equipments and manpower. A hospital becomes an efficient system when all its activity components (diagnosis, treatment and management) can be linked to provide a better service to its patients.
However, there exist intense interrelationships among the entities to achieve the objective of each activity component. It can be observed that the admission, diagnosis, operation and staying of a patient in a hospital requires a diverse interaction with various entities such as physician, matron, pathologist, surgeon, nurse and accountant in terms of information flow. Most importantly the entities mentioned need to interact with each other to provide efficient service to the patient. In this way, a complex interrelationship among the entities emerges and hence the hospital system becomes complex. Since the information related to all the activities of this complex system is difficult to manage in a manual way, there is a risk of mismanagement and inefficiency within the system. Hence, the efficient service to the patient is hampered and hence, the objective of the hospital is seriously blocked.

A Computer-based Information System (CBIS), which is used to store, capture, retrieve, manipulate, analyse and model all kinds of information can be considered as an appropriate solution to increase the management efficiency of a hospital. In addition, the modelling and analytical ability of the CBIS could provide a better way to understand the interrelationships among the entities of the hospital. This, in turn, can be used to develop a way to use the manpower in an effective way.

A robust approach is necessary to develop an efficient CBIS/software system. Software Development Life Cycle (SDLC) can be considered such an approach consisting of a number of phases including requirement specification, system specification, system design, programming and coding, testing and maintenance. Among the phases software requirement specifications can be considered as the most important because the reliability, efficiency, usability and maintainability of a software system mainly depends on this phase [5]. This is especially true when the CBIS for a large complex system like hospital to be developed. Inappropriate capturing of requirement specification will cost a large amount of money in the operational environment of a software system. Therefore, the paper presents the capturing procedures of requirement specification for a CBIS to be used to support the activities (diagnosis, treatment and management) of a hospital. The following section will give an overview of the methodology be adopted in developing the requirement specifications. This is followed by the presentation of the results obtained.

2. Methodology for Requirement Specifications
There exist a number of methods to specify software requirements. They can be categorised as partitioning, abstraction and projection [1]. Computational models such as Entity Relationship Model, Object Model are mainly adopted in partitioning and abstraction approaches respectively. On the other hand, projection is mainly based on the analysis of the system problem domain. The disadvantage of partitioning and abstraction is that they are very much developer-oriented and therefore, unable to capture the essence of software requirement specifications. In addition, users don't want to specify their requirement specifications in terms of entity or object.
The projection-based approach puts emphasis on the understanding of the system problem domain and hence, enables a transferring of information from the user-domain to analyst domain. Thus, an appropriate capturing of information from the user-domain can be ensured. This information through a series of processes can be transformed into more concrete software requirement specifications. Therefore, the paper considers a domain-based approach to develop requirement specifications for the CBIS to be used in a hospital. Problem domain knowledge will be transformed into requirement specifications through the analysis of its environmental, behavioural and solution characteristics.

3. System Problem Domain
A system problem domain consists of the identification of the areas in which the CBIS will be applied [2]. The main components of a hospital system can be considered as diagnosis, treatment and management and interrelationships exist among theme as mentioned previously. This viewpoint has been captured in the CBIS problem domain as illustrated in Figure 1.

![Figure 1: System Problem Domain](image-url)
It can be seen from Figure 1 that each component consists of a number of elements and they are again dependent with each other. For example, 'Diagnosis' component consists of 'Sampling Activity', 'Testing Activity' and 'Report Factors'. A testing activity cannot be carried out without performing sampling activity. The testing activity should focus on the factors to be included in a report. Thus, it can be observed that the elements of the 'diagnosis' component are dependent with each other.

The main components are also dependent with each other and this can be seen from Figure 1. The entities associated with each component element are also captured. For example, Sampling Activity can only be performed if there exist data on patient profile, test type (x-ray) and physician's reference. The detailed information on the identified component elements and their entities require a through examination taking domain characteristics into account. This in turn will work as a basis to define the system requirements for the CBIS and will be presented in the following section.

4. System Problem Domain Characteristics
The problem domain of a system consists of three characteristics. They are known as environmental, behavioural and solution characteristics [3]. The section presents the characteristics taking the system problem domain into account.

4.1. Environmental Characteristics
This is concerned with recording the general information that characterizes the domain- the interested users or stakeholders in the domain, the fundamental nature of the problem, and the general environmental factors and constraints.

4.1.1 Stakeholder List
Information on status, function, objective, dependency with others and problem faced are recorded for each of stakeholder as shown in Table 1. The status will give information on the type of the stakeholder. Information on stakeholder's objective and function will assist the system developer to identify system events. Information on dependency will help analyst to understand relationship among the users of the system. Identification of the problem works as a basis to identify user requirements.

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Status</th>
<th>Function (i.e. jobs tasks)</th>
<th>Objective (i.e. why?)</th>
<th>Associate With other Users</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receptionist</td>
<td>Primary</td>
<td>- Registration - Admission - Report collection &amp; delivery - Patient query</td>
<td>- Preserving previous records - Obtaining patient information</td>
<td>- Account officer - Pathologist - Matron - Medical officer</td>
<td>- Missing case history - patient can not remember or preserve his previous record</td>
</tr>
</tbody>
</table>
4.1.2. Environmental Factors and Constraints

Information on environmental factors and constraints has been captured, which will help to identify the non-functional system requirements. As an example, the system should not accept redundant data.

4.1.3. Problem List

The list provides a scope to model the real world problem by identifying its symptoms, impact on organisation, cause and priority. The missing case history of a patient can be considered as an example of such a real world problem. The symptoms of this problem is recognised when a patient comes more than once in the hospital and the person concerned is unable to remember it. Consequently, same thing needs to be recorded again and again resulting increasing overhead cost in hospital. The cause of this problem is the presence of manual system, which is unable to sort the information in a short period of time. Therefore, an automated solution to this problem required to be given the top priority. The above is illustrated in Table 2. In this way, the information on the other problems has been captured.

<table>
<thead>
<tr>
<th>Name</th>
<th>Symptoms (who or what are affected)</th>
<th>Impact on organization</th>
<th>Cause of Problem</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing case history</td>
<td>- When patient come more than once, they need previous records</td>
<td>- Overhead cost</td>
<td>Register book is large or may have many volumes and hence, difficult to maintain</td>
<td>1</td>
</tr>
</tbody>
</table>

4.2. Behavioural Characteristics

This is concerned with an understanding of the system components consisting object, process and event. The information on the objects enables an understanding of the static aspect of the system while the information on process and event facilitates the understanding of dynamic aspect.

4.2.1. Object Inventory

Anything of interest in the system problem domain can be considered as object [3]. The information captured for each object include its definition, its dependency with other objects, its access and availability, its state in the problem domain and its associated process. For example, the object 'Patient Registration Form' (as illustrated in Table 3) has dependency with other objects such as nurse, physician and accountant. The capturing of this information will help to develop conceptual data model using ER model. Patient's profile information is located in a registry book. This could help to identify the attributes associated with this object. Information on the state of the problem will help to identify the state of the object in its domain. For example, a patient may be discharged or stayed at the hospital. How the state of the 'Patient Registration Form' will be changed can be understood by identifying the processes.
Table 3: Object Inventory List

<table>
<thead>
<tr>
<th>Object name</th>
<th>Description</th>
<th>Dependencies/Associations with other Objects</th>
<th>Access, availability and Location</th>
<th>Possible States</th>
<th>Processed by</th>
</tr>
</thead>
</table>
| Patient Registration Form | Who is admitted or registered in the hospital for long or short-term treatment. | - Doctor  
- Nurse  
- Receptionist  
- Account officer | Patient information is maintained in a registry book. | - Patient Registered  
- Patient discharged | - Patient Registration  
- Patient Admission  
- Patient Discharging |

4.2.2. Process

Process is an activity, which operates on an object and changes its state. The process 'Patient Registration' as shown in Table 4 will act on the object 'Patient Registration Form' and will change its value. For example, when new patient get registered in a hospital, the number of patients will be changed. This implies that through the 'Patient Registration' process the value/state of one of the attributes (for example number of patient) of the 'Patient Registration Form' has been changed. This represents the dynamic aspect of the system. Therefore, the information on the objects associated with a process is to be captured. The recording of information on the other factors such as user, function and event will be discussed eventually.

Table 4: Process List

<table>
<thead>
<tr>
<th>Process name</th>
<th>Part of function…</th>
<th>Description</th>
<th>Users</th>
<th>Objects</th>
<th>Start/Stop Event</th>
</tr>
</thead>
</table>
| Patient Registration | Monitoring Patient | Patient profile entry | Receptionist | Patient registration form | Start: Request for patient registration  
End: Registered |

4.2.3. Event List

Event is an instantaneous point in time where something of interest occurs. Every process is associated with at least two events. For example as shown in Table 4, the event 'Request for Patient Registration' will trigger the process 'Patient Registration' and the event 'Registered' will stop the patient registration process. Therefore, an event causes something of interest, which is in this case the triggering, and stopping of 'Patient Registration' process.

4.3 Solution Characteristics

Solution characteristics is concerned with the recording of general information on the solution to a problem. This consists of identification of information generation requirements and system function outline.
4.3.1 Information Generation Requirements

The success of a system depends on the generation of appropriate information from the system. A review of the results presented in Table 5 shows that 'Patient Registration' process should generate the information as described in the 'content' column, otherwise the applicability of the function will not carry any weight to its user.

<table>
<thead>
<tr>
<th>Report Name</th>
<th>Generated by process</th>
<th>For user</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient Registration Details</td>
<td>Patients registration</td>
<td>Receptionist</td>
<td>This report contains all information about patients such as ID, name, age, sex, father/husband name, address, and reference.</td>
</tr>
</tbody>
</table>

4.3.2. System Function Outline

A function consists of a number of processes. System functions are used to define the overall boundary of the system. The major functions required to address the problem components (diagnosis, treatment and management) of the system problem domain have been identified. Therefore, this enables the identification of the system boundary because the functions facilitate setting the overall activities of the proposed system. A description of the major functions is presented below.

- 'Disease Detection', associated with 'Diagnosis' problem component [Figure 1] for analyse disease and generate test report
- 'Monitoring Patient', associated with 'Treatment' problem component, to support medical and surgical treatment procedures
- 'Decision-making', associated with 'Management' problem component, to support measure to increase the service efficiency of the hospital

Table 7 illustrates the 'Monitoring Patient' function taking its objectives, associated processes and the problem it addresses.

<table>
<thead>
<tr>
<th>Function Reference</th>
<th>Objectives</th>
<th>Associated Processes</th>
<th>Problem addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring Patient</td>
<td>Preserving all information of patient to provide service</td>
<td>- Patient registration - Rostering - Booking</td>
<td>Support medical and surgical treatment procedures</td>
</tr>
</tbody>
</table>
5. System Architecture
The behavioural (objects, processes and event) and solution aspects (outputs and functions) of the system have been identified. These aspects can be considered in order to identify the basic components (including input, process and output) of a system and thus, provide a means to develop its basic architecture. This is shown in Figure 2, which demonstrates the essential architecture for the system following a consideration of the data, functional processes and output from the system. This is also considered to be the basic constituent for developing an information system.

![Figure 2: System Architecture](image)

Data are the main components in building any system. Based on the object inventory schedule, the data required for the CBIS can be categorised as:

- Patient's profile including name, address, age, sex, id, type, father/husband name, reference, case history
- Administrative data including staff's profile, financial, patient booking
- Treatment data including disease type, medication, daily patient monitoring report, anaesthesia, OT details, surgeon profile
- Diagnostic data including test type, test code, sample receiving and delivery data, report generation date, equipments, pathologist profile

The functions included in Figure 2 considering the system function outline [Section 4.3.2]. The main outputs from the system can be considered as: 1) statistical data including number of patient admitted in a month, medicine inventory, medicine selling amount, number of discharged patient in a month, 2) graphical display of the statistical data and 3) generation of various decision scenarios considering system generation requirements [Section 4.3.1].

6. Discussions and Conclusion
Environmental, behavioural and solution characteristics of the problem domain of the system [Figure 1] have been captured in a structured way in order to specify its
requirements. Consequently, it allows all the aspects of the system to be captured and its boundary to be defined. The identification of the architecture for the CBIS could provide a means to obtain an initial idea about its data, processes and output. In conjunction with requirement specifications this could provide a basis to proceed to the next stage of system development, which are system specification and design. For example, using object inventory schedule [Section 4.1.1] an Entity-Relationship Model can be developed to specify the data for the system. In the same way, using process inventory schedule [Section 4.2.2] the Data Flow Diagrams (DFD) can be constructed to facilitate structured analysis of the system. This will in turn help to perform the structured design of the system using structure charts. State transition diagram can be developed with the help of event schedule [Section 4.2.3].

There exist other approaches to develop software requirement specifications such as Object-oriented Modelling Technique [4] and VORD method [6]. The approaches mainly put emphasis on developing the system models (including object, dynamic and functional models) during requirement specifications phase of the software development life cycle. These system models based on computational concepts rather than application domain concepts. Therefore, using computational concepts, it is difficult to transferring the application domain knowledge from the domain to the analyst. On the other hand, application domain based approach facilitates a transferring of the knowledge from domain to analyst since it considers an intensive analysis of environmental, behavioural and solution characteristics of the domain. Hence, this paper considers a domain-based approach to capture software requirement specifications.
References:


