Volume 2, No. 2, Mar-Apr 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Information Retrieval System of Hospital using UML and Data Cube

Rajesh Kumar
School of Mathematics and Computer Applications
Thapar University
Patiala, India
rakumar@thapar.edu

Rajni Aron*
Computer Science and Engineering Department
Thapar University
Patiala, India
rajni@thapar.edu

Abstract: UML is important for software analysis and designing. Each UML diagram depicts every relationship between model and reality. The technique for making a model of our ideas is a use of Abstraction. The UML abstractions are developed as conventions to be learned and used easily. Data Cube is used under different abstraction levels in a dimension hierarchy. This paper describes how information can be retrieved through UML and Data Cube. UML activity diagram, sequence diagram and class diagram have been designed to retrieve the information efficiently. It also proposes the framework of the hospital system. This paper contains how we can use Data Cube for fast retrieval of information and how several queries can be performed. It also advocates a wider use of Data Cube for solving problem.

Keywords: UML, Data Cube and Hospital System

I. INTRODUCTION

Conceptual models are formal descriptions of application domains that are used in early stages of system development to support requirements analysis. The Unified Modeling Language was formed by integrating several diagramming techniques for the purpose of software specification, design, construction and maintenance.

The Unified Modeling Language (UML) has quickly become the de-facto standard for building Object-Oriented software [1]. UML is a graphical notation for expressing object-oriented designs. It is called a modeling language and not a design notation. UML allows representing various aspects of the system, not just design that has to be implemented. Due to the ability of UML to create different models, it has become an aid for understanding the system, designing the system, as well as a notation for representing design. The design activity begins when the requirements document for the software to be developed is available and the architecture has been designed. In the present paper UML model is proposed for information retrieval of hospital and this model helps to extract the information of hospital from the UML model. A real case study of Rajindra government hospital, Patiala, India is considered. Data Cube are designed for fast retrieval information and using SQL queries. Data cube use to represent N-dimensional using 2-D relations [6].

Data cube computes the aggregate information within the database. It provides reliable vision for robotics, medical imaging, surveillance, inspection, teleconferencing, animation etc [2, 3, 4].

II. UML MODEL FOR INFORMATION RETERIVAL SYSTEM OF HOSPITAL

The modeling part of the system is essential for designing purpose. UML is used to define a software system. To glean the useful information early in information retrieval system of hospital, Here we have designed three UML diagrams Activity Diagram, Sequence diagram and class diagram.

A. Activity Diagram

Activity Diagram depicts behavioural features of a system or business process [5]. Activity diagram is the object oriented concept of flow charts and data flow diagrams and used to explore the logic of complex operation, software processes etc. Although activity diagrams can potentially model the internal logic of a complex operations. An activity diagram enables us in the process to understand and monitor the development of system functionality.

Figure 1 shows the activity diagram of information retrieval system of hospital. In the hospital system, we have taken two types of patient, indoor patient and outdoor patient. In case of indoor patient, OPD registration center will check availability of the bed. If bed is available then patient will be registered. As shown in Figure 1, we will use fork to create concurrent activities. Activity update record patient will activate after the allotment of bed. Patient will consult with the doctor. Doctor will prescribe the medicines and tests. After that treatment will start. Staff looks after the patient. As soon as the patient is fine, patient will pay all the bills. Patient will leave the hospital. Activity diagram will be terminated. If outdoor patient arrives for registrations, patient will take the appointment from the doctor. After taking the appointment, patient will consult with doctor. Doctor will prescribe the medicines and tests. Staff looks after the patient. Update record patient activity will update the record of the patient. Outdoor patient will pay the fee of the doctor and medicine bill. We will merge two activities like patient bill and update patient record. Patient will leave the hospital and activity diagram will be terminated.

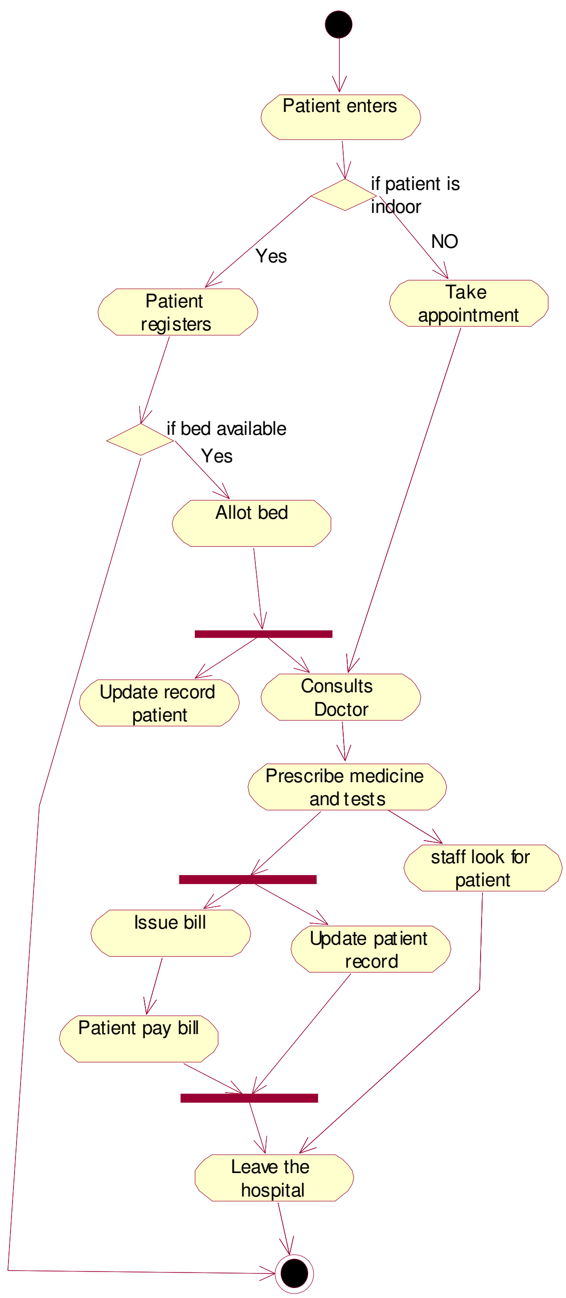

Figure 1: Activity Diagram

B. Sequence Diagram

A sequence diagram is a graphical view of a scenario that show object. A sequence diagram is used to show the behavior sequence of a use case. A sequence diagram shows a set of messages arranged in time sequence. In the sequence diagram, we have shown the active object those are active in processing and lifeline shows object instance that participate in the sequence being modeled. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior, which can be difficult to extract from static diagrams or specifications. UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objects or components of a system. Sequence diagrams can be used to explore the logic of a complex operation, function, or procedure.

The sequence diagram also shows the sequence of transfer of the patient from one object to another object. The sequence diagram is shown in Figure 2. There are nine objects in the sequence diagram i.e. Indoor patient, Registration, Ward, Outdoor patient, Doctor, Patient record, Staff, Bill and Expenditure. The transportation between the objects is represented by an arrow and message on that arrow. The vertical line shows the lifeline of the objects. Lifeline represents the role over time through the entire interaction. Messages are shown as arrow between lifelines.

Firstly, indoor patient registers and registration department will check the availability of bed in the ward. After getting the information from the ward section, registration department will allot bed to the patient. Registration department will provide the bed information and patient information to patient record section.

Outdoor patient will take appointment from the doctor. After getting the appointment from doctor, patient will consult with the doctor. Doctor prescribes the medicine. Patient will pay the fee. Outdoor patient will leave the hospital.

Indoor patient will consult with doctor. Doctor prescribes the test and medicine to the indoor patient. Doctor will give the information to patient record section. Staff looks for the patient. Patient record section will give all information of the patient to the bill section. Bill section will check all the details regarding patient bill, charges etc. Bill section will issue the bill. Indoor patient will pay all the bills. As soon as patient is fine, doctor will discharge the patient. Staff asks for salary from expenditure department and that department will pay the salary.

C. Class Diagam

A class diagram shows the existence of classes and their relationships in the logical view of the system. A class diagram shows the graphical representation of the static view. The class diagram can be used by analysts, business modelers, developers, researchers and testers throughout the software development life cycle. Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view. In the current style of the object-oriented programming, classes are highly visible but their relations (associations, data structures, design patterns) are buried inside class definitions and not easy to find. UML gives us one unified view where classes and their relations have the same importance [7].

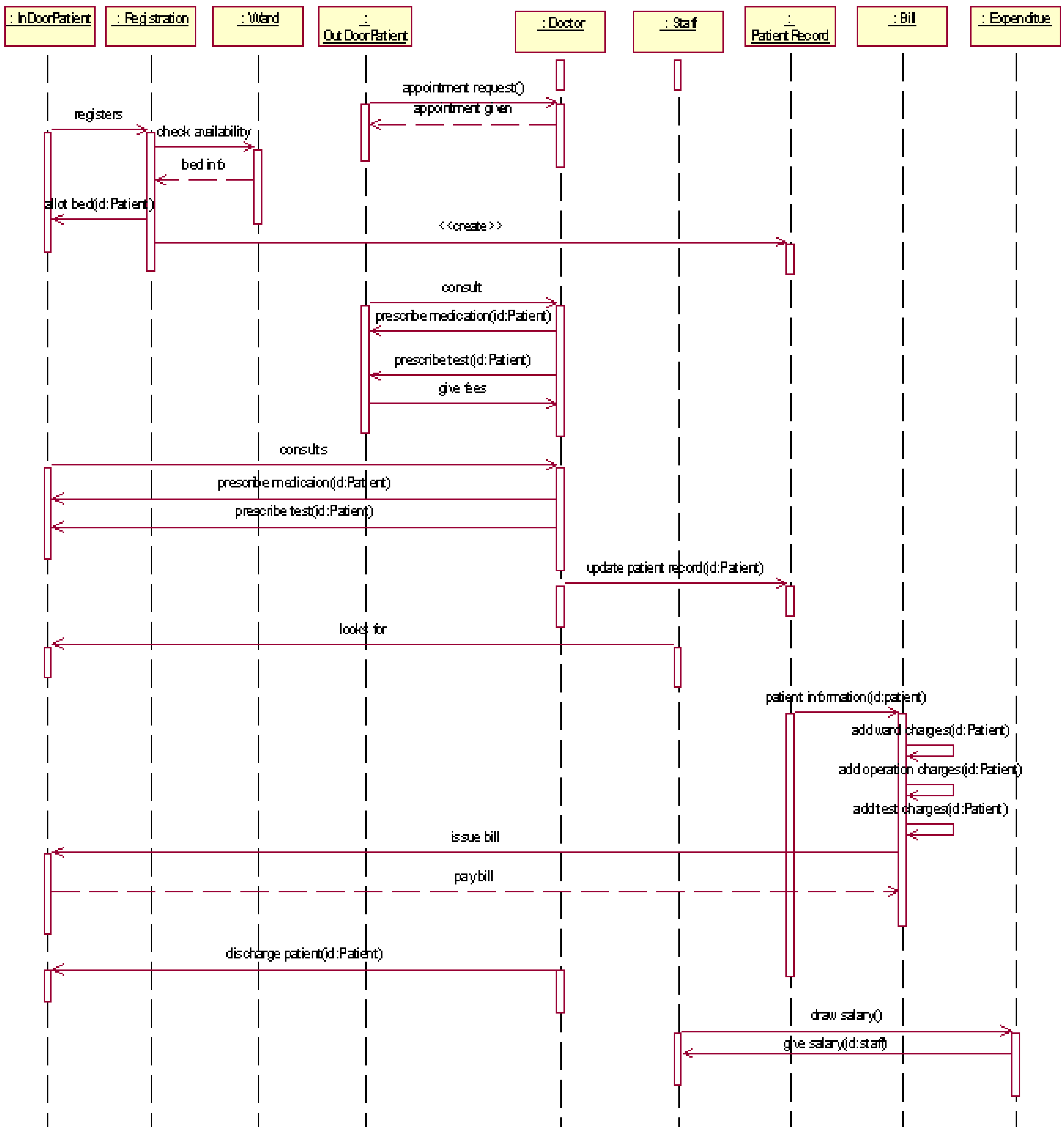

Figure 2: Sequence Diagram

C. Class Diagam

A class diagram shows the existence of classes and their relationships in the logical view of the system. A class diagram shows the graphical representation of the static view. The class diagram can be used by analysts, business modelers, developers, researchers and testers throughout the software development life cycle. Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view. In the current style of the object-oriented programming, classes are highly visible but their relations (associations, data structures, design patterns) are buried inside class definitions and not easy to find. UML gives us one unified view where classes and their relations have the same importance [7].

We have modeled our system with the help of UML class diagram. The system contains fourteen major classes Outdoor patient, Indoor patient, Expenditure, Staff, Patient, Department, Doctor, Registration, Ward, General ward, Special ward, Hospital record system, Bill and Patient record with attributes and operations. Indoor patient class and outdoor patient class inherit the feature of the patient class. In patient class; patient_id, name, sex, phone-number, addresses, D.O.B are attributes and consult doctor () is an operation.

Outdoor patient has same attributes like patient class and appointment request (t: time) is an operation. Outdoor patient take the appointment from department to consult the doctor. Outdoor patient and department have relationship. Department has one relation with outdoor patient and outdoor patient has multiple relationships with department because department has many outdoor patient at time but outdoor patient will take appointment from one department at a time.

Guardian/relative name, guardian/relative phone number are attributes of indoor patient class. Request blood (blood group: string, id: patient) is an operation of indoor patient class. Indoor patient comes and registers. Indoor patient will pay the bill, after payment of the bill, patient will be given room according to availability and choice of patient. Staff takes care of the patient. Doctor will see the report of indoor patient, kept the record own self. As soon as patient is fine, the doctor will discharge the patient. All types of relationships, attribute and operation have shown in the class diagram given in figure 8.

Allot bed (id: patient) is the operation of registration class. Registration class will check the availability of ward.

Get name (bed no: integer), availability () are operations of ward class. General ward class and special rooms class inherit the feature from ward class. Ward no. and bed no. are attributes of general ward class, bed status () is an operation. Room no is the attribute of special rooms class and room status () is an operation. Id, name, sex, phone-number, address and D.O.B are attributes of doctor class. Prescribe medicine (p: patient), prescribe tests (p: patient), login (id: integer, password: string), review patient record (pr: patient record) and discharge patient () are operations of doctor class. Doctor has one relationship for patient record class but patient record has many relations for doctor class. Doctor updates the patient record.

Staff class generalizes from doctor class. Id, name, sex, phone-number, address, D.O.B are attributes of staff class. Take care of patient (Pid: Patient), login (id: integer, password: string), draw salary () are operations of staff class. Staff has one to many relationships for patient class and patient has many relations to staff class because one patient looks after by one staff member or more than one but one staff member looks after zero patient, one patient or more at a time. Dept_id, patient_id, dept_name are attributes of department class and add () is operation. Each department works for doctor. Department class depends on the patient class. Patient will be alloted one department at a time but one department has many patients at a time.

In Patient record class; id, disease, treatment, date entry, date of discharge, date are attributes Update patient record (id: patient, date: date) is an operation of patient record class. Patient record class will give information to bill class. Patient record class will view and update the patient record files those are sent by doctor. Bill no, billamt and medicine info are attributes of bill class. Get paid info (id: patient), add test charges (id: patient), add ward charges (id: patient) and add opercharges (id: patient) are operations of bill class. Bill has one to many relationships to patient class but patient class has one relation to bill class because bill class will send bill to one patient at a time but patient will pay the many bills at a time.

Hospital consists of two live subsystems, OPD and Nursing Home. Under hospital management first we analyze the working of both subsystems and then automate the system. OPD Patients come into the hospital by taking appointment on the phone or directly. Sometimes emergency case can also be there. Each patient in OPD is given time for 5 to 10 minutes on an average. Medicines are suggested to the patients, which can be collected from any medical shop outside the hospital. In some cases medicine/ injection can be given to patient in the hospital as well. Some tests can be suggested to the patients for which laboratory is referred.

Patients pay fees to the bill section. (Fees that include doctor fees plus any charges related to injection, indoor medicine etc). Doctor himself keeps the record of outdoor patients in OPD register. Patients in OPD are told to take the OPD slips given to them and should bring the same when they come to see doctor the next time.

There are general ward and special rooms in the nursing home as shown in Figure 3. Whenever any patient enters in the nursing home, details are entered in the register (date of admit, name, age disease, village) and when he/she is discharged, the date of discharge is written in the register. If a person discharges under normal conditions then a tick mark is put after the discharge, if patient is not under normal conditions and the doctor feels that he cannot treat the patient well, then patient is referred to another doctor and the name of referred doctor is written after date of discharge in place of a tick. If patient dies then ‘dead’ is written in place of a tick. On discharging the patient his / her file is referred in the Patient record class, date of admission and date of discharge are included along with the bill information and any treatment suggested after discharge is also written.

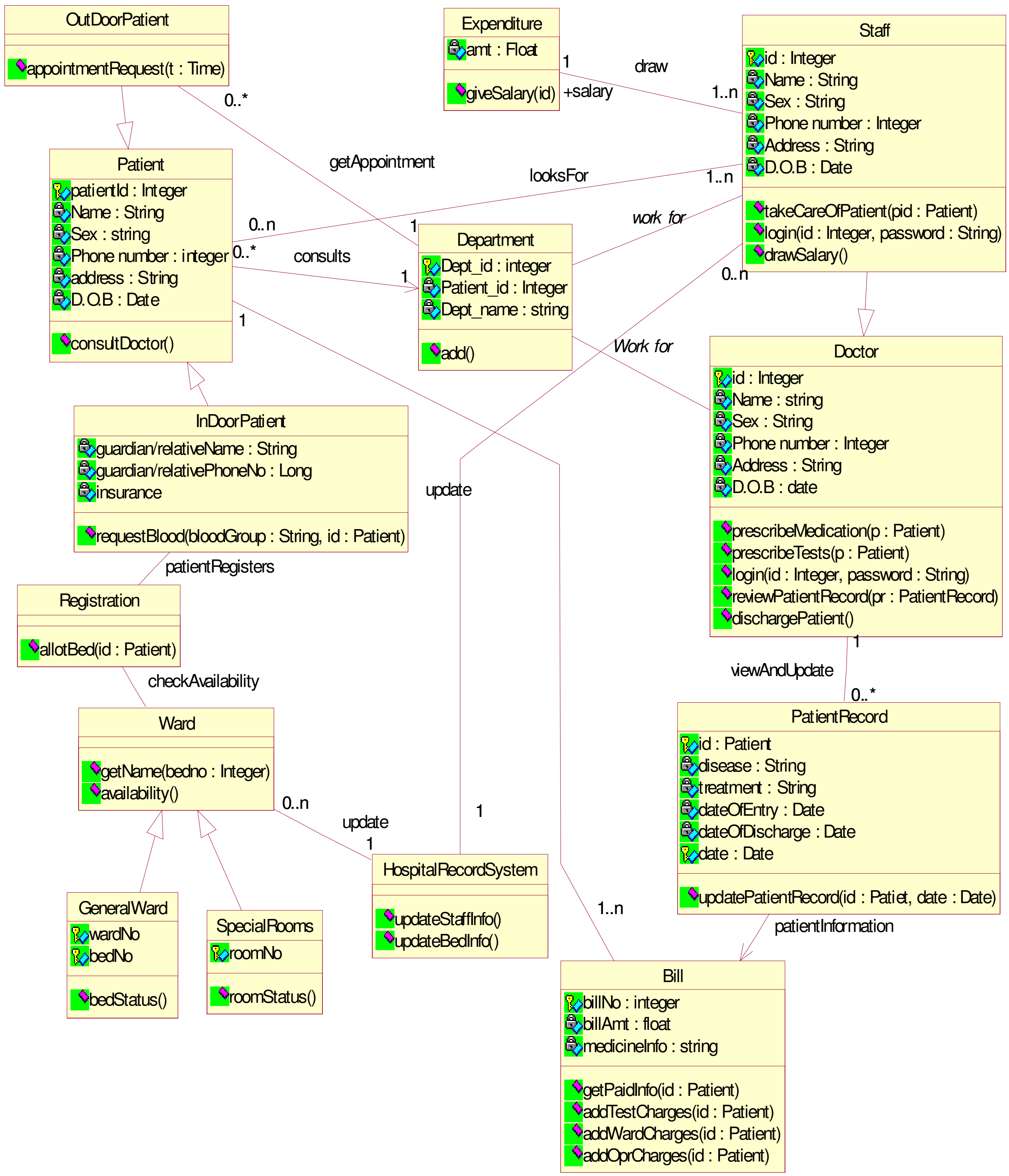

Figure 3: Class diagram

III. DATA CUBE REPRESENTATION

Data Cube queries compute aggregates over database relations at a variety of granularities and they constitute an important class for decision support queries [8, 9]. The UML class model can be implemented through the design of Data Cube by the three dimensional representation of Data Cube of information retrieval system of the hospital. We generally consider the attributes mainly department_id, doctor_id, patient_id, patient_disease and doctor_specialization. But in Data Cube we will take three attributes Department_id, Patient_id and Doctor_id.

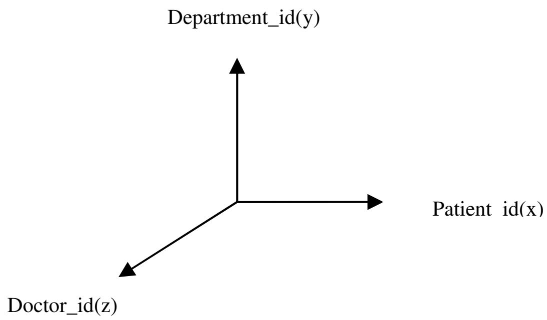

Figure 4: 3-dimension view

Figure 5 depicts a small, practical Data Cube example of the Hospital. This particular Data Cube has three feature attributes --- Department_id, Doctor_id and Patient_id. We have described the Data Cube as a conceptual model. This is certainly true. However, in the case of a MOLAP server, it is also the physical model, because MOLAP stores the cube structure directly as a multi-dimensional array. Conversely, ROLAP server must map this representation to a relational design. The content of each cell is the count of the number of the times that specific value comes collectively in the database. The main goal to design Data Cube is to extract the information in a faster way.

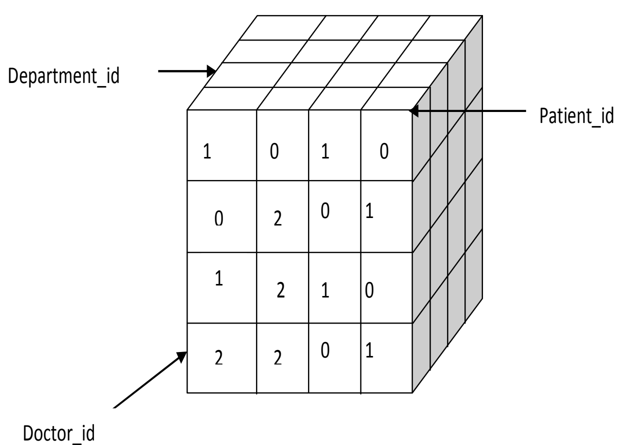

Figure 5: A simple view of Data Cube

Query 1.
Select doctor_id, dept_id from his_info group by cube (doctor_id, dept_id);
The output of the above query shows possible combinations of dept_id and doctor_id.

DOCTOR_ID	DEPT_ID
1112	
2112	
2113	
3111	
3112	
5111	
5112	
6111	
6112	
8111	

DOCTOR_ID	DEPT_ID
8112	
9111	
	1010
1112	1010
9111	1010
	1011
2112	1011
3112	1011
	1012
2113	1012
	2020

DOCTOR_ID	DEPT_ID
8111	2020
	3030
6112	3030
	4040
5112	4040
	5050
5111	5050
	7070
3111	7070
	8080
6111	8080

DOCTOR_ID	DEPT_ID
	9090
8112	9090

35 rows selected.

Query 2. Select doctor_id, dept_id, count (*) from his_info group by cube (doctor_id, dept_id) having doctor_id='8112';

DOCTOR_ID	DEPT_ID	COUNT (*)
8112	9090	2
8112		2

Table I.

Department_id	Doctor_id	Patient_id	Doctor_specialization	Patient_disease
1010	1112	3007	Medicine	Pheumonia
8080	6111	3026	Urology	Kidney Stone
1012	2111	3024	E.N.T	Malignant neoplasm's of lips
1010	9111	3019	Medicine	Tuberculosis
7070	3111	3017	Neurology	Malaria
2020	8111	3012	Surgery	Cancer
3030	6112	3002	Orthopedics	Bone tumor
9090	8112	3004	Cardiology	Heart Problem
2020	8111	3016	Surgery	Hernia
9090	8112	3014	Cardiology	Angina Pectoris
5050	5111	3006	Skin& V.D	Ring worm
4040	5112	3023	Pediatric	Fever
1011	2112	3005	Gynecology	Preterm labour
1011	3112	3015	Gynecology	Gestational Diabetes

IV. CONCLUSION & FUTURE SCOPE

We conclude by saying that UML modelling is a very efficient modelling language to represent such system. By UML activity diagram, sequence diagram and class diagram, we are able to model the whole system in an efficient way. By implementing data in Data Cube, we can get any information very quickly. We can perform a number of queries very rapidly. We can get the desired result in the form of any record of the patient and doctor by using less number of join. We can retrieve information only through one table. We can handle query for instant decision of bed allocation for patients, and request for the bed transfer easily. Data Cube hides the information at a backhand level. For nonexpert UML or database users, the cube class graphical notations facilitates the definition of initial user requirements.

V. REFERENCES

[1] J. W. Dzidek, E. Arisholm and C. L. Briand, "A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance", IEEE Trans. Softw. Eng. Vol. 34, No. 3. pp. 407-432. May 2008.

[2] J. Han and M. Kamber, "Data mining concepts and techniques", an imprint of Elsevier Morgan kuafmann publishers, second edition, March 2006.

[3] A. Ivanova and B. Rachev, "Multi dimensional models – Constructing DATA CUBE", Proc. of CompSysTech'04**,** Rousse (Bulgaria), 2004.

[4] R. Missaoui, C. Goutte, A. K. Choupo and A. Boujenoui, "A probabilistic model for Data Cube compression and query approximation" In Proceedings of the ACM Tenth international Workshop on Data Warehousing and OLAP, DOLAP '07. ACM, New York, NY, pp.33-40, November, 2007.

[5] J. Rumbaugh, I. Jacobson and G. booch, "The United Modeling Language Reference Manual" published by Pearson Education Asia Pte. Ltd. (Addison-Wesley Object Technology Series, second edition, 1998.

[6] S. Palaniappan, L.S. Chua, "Clinical Decision Support using OLAP with data mining" in IJCSNS Internatinal journal of computer science and network security, VOL.8, pp.9, September 2008.

[7] M. Soukup and J. Soukup, "The popularity cycle of graphical tools, UML, and libraries of associations", In Companion To the 22nd ACM SIGPLAN Conference on Object-Oriented Programming Systems and Applications Companion (Montreal, Quebec, Canada, OOPSLA '07. ACM, New York, NY, pp. 753-756, October 21 - 25, 2007.

[8] A. K. Ross and A.K. Zaman, "Optimizing Selection over Data Cubes" , In proceeding of the IEEE international conference on scienti and statistical database management, November 24, 1998.

[9] A.K. Ross and D. Srivastava, "Fast Computation of Sparse Datacubes", In Proc. of the 23rd international Conference on Very Large Data Bases. M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, and M. A. Jeusfeld, Eds. Very Large Data Bases. Morgan Kaufmann Publishers, San Francisco, CA, pp. 116-125, August 25 - 29, 1997.

