Department Of Computer Science

ETL (Extraction, Transformation and Load) & OLAP (Online Analytical Processing) Reporting
AND
Complex Obs Support
For
(OpenMRS)Open Medical Record System
Interim Report

Date: 9th May 2008

Contents

1 Review:

1.1 Introduction:

In today's world where medical science is reaching new heights day by day, number of things are being invented to fight against with deadly diseases, billions of pounds are being invested for research and creating new medicines about 40 million people are infected by or dying with HIV/AIDS and which are mostly in developing countries. To prevent or treatment of this disease in developing countries is quite difficult task due various reasons like lack of man power, unavailability of efficient management system, time constraints and money. Because of these reasons most of the programs running in these countries manage large information with simple spreadsheets, badly designed database or flat files. All of these functionality results in loss of various resources, to come over with this problem we need investigate an approach not only at management level but also try to reduce unwanted and duplicative efforts.

To overcome with these difficulties, Open Medical Record System (OpenMRS) which was formed in 2004 which is open source framework for developing countries. OpenMRS is non profit organization under guidance of Regenstrief Institute, Inc which is world leading medical research institute along with another medicine based organization. The aim of OpenMRS is to implement medical record system which will help implementers at on site with efficient easy to access database of records. OpenMRS has been implemented to various countries of Africa such as South Africa, Kenya, Rwanda, Lesotho, Zimbabwe, Mozambique, Uganda, and Tanzania. Providing efficient management tools is not only goal the OpenMRS team but also to reduces unwanted and reworking efforts. It is an effort of building common platform on which medical informatics can be built.

The OpenMRS system is conceptual view which is independent of types of information gathered and it can be customized for many different purposes. Making maximum use of coded information so it can use used for summarization and analysis is what the principle of OpenMRS. All the gathered information such as different kind of diagnosis, various tests, procedures, drugs, medicines, questions and answers with patient is what called as a concept directory which is at heart of OpenMRS system. OpenMRS is client server application where many client applications will access information stored at server and make use of that information to implement proper functionality. Following is figure of OpenMRS architecture.

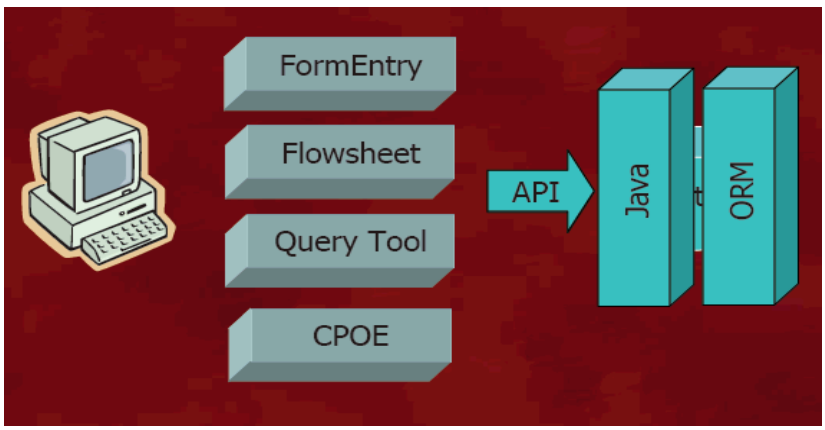


Figure 1: OpenMRS architecture.

[source: OpenMRS An EMR for Developing Countries Burke Mamlin, MD Paul Biondich, MD MS …and the OpenMRS collaborative page page no 24]

1.2 Aims and Objectives

1.2.1 Module1: Complex Obs Support

Current implementation of OpenMRS supports simple clinical observations like hemoglobin test, patient's weight, blood test etc. and loads these observations into database. But clinical observations are not always simple and they can become complex like X-ray images, genetic sequences, MRI-Scan etc. The aim of the project is to extend current implementation of OpenMRS to support these complex observations found during clinical setup. The proposed data model of OpenMRS deals with this issue with help of complex_obs and mime_type tables. Project objective is extend Current API implementation of OpenMRS to deal with processing of these complex observations and loading data into these tables. By making use of the XML files or any other file formats it can be possible to deal with complex observations and loading data into database(complex_obs table), but system also needs to cope when total amount of data is huge or reference to another file system or database is given via URN or URL.

1.2.2 Module 2 : Reporting for ETL/OLAP

The goal of this project is to developing ETL functionality in order to support OLAP analysis. OpenMRS has already have Data Model which consist of various tables. We need to first develop star schema on basis of which Data Warehouse will be created. Also We need to write create ETL script which will load data from OpenMRS data model to Data Warehouse which will be create. We will need to look on reporting side which will be generated based on data present in Data Warehouse.

2 Background:

2.1 Module1: Complex Obs support

2.1.1 Obs table primer

Observation:

An observation is measurement of any finding or observation in clinical scenario. We can say that any findings like if we do a hemoglobin test on patient then hemoglobin level of patient, weight of patient, any question ask to patient regarding medical examination or any other physical findings during medical checking. In short any data or information collected during clinical setup can be considered as a observation.

Concept:

A concept is way by which we can describe an observation we have found in our system. If we want to capture information in the data repository it can done easily with the help of following table which is part of data model describe by OpenMRS team.

Any kind of observation found during clinical setup generally stored in this table as one row. Following are details about some of the fields of this table.

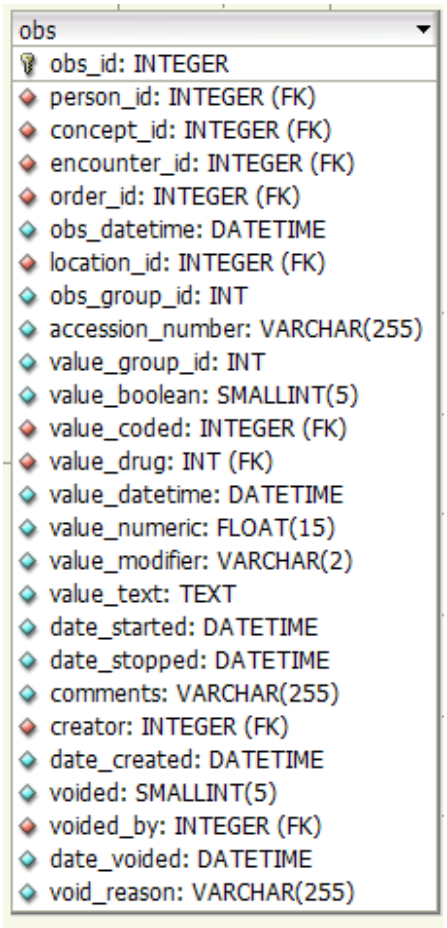


Figure 2 : obs table.

[source: http://openmrs.org/images/5/59/Openmrs_data_model_1.10.png OpenMRS Data Model access on 20rd april]

- obs_id: It is unique identifier for any given row in the table.
- patient_id: Its foreign key for for "obs" table and primary key in "patient" and it refers to internal patient identifier number.
- term_id: This another foreign key which is primary key for "term" table. It refers to the concept which describes what is being collected by system.
- location_id: It is primary key for "location" table which is being referred here as a foreign key. It indicates where clinical observation found and it will be typically either clinical setup or laboratory.
- encounter_id: It refers to particular visit of any patient while clinical observation.

For e.g.,
Patient: ABC
MRN: 99TU-2, which corresponds to patient_id: 999
Clinic Vist Date: 9/10/07 @ 3.40p.m.
Location: XYZ Medical Center
Hemoglobin: 11
Weight: 40 kg
Date of TB treatment: 7/1/2004

Gastrointestinal Exam: Hepatomegaly.
The data which we have found in above example is quite straight forward and can be easily loaded into database.

Obs_id	pat_id	trm_id	lctn_id	encntr_id	sub_id	val_bool	val_cod	val_dt	val_num	num_mod	val_txt	d_e_t	enterer	comment
1	999	15	4	567					11.5			9/11/2007	4	
2	999	3645	4	567					45			9/11/2007	4	
3	999	1146	4	567				1/1/2002				9/11/2007	4	
4	999	1196	4	567			5008					9/11/2007	4	

While storing and retrieval of the images the tables which we need to into consideration are follows:

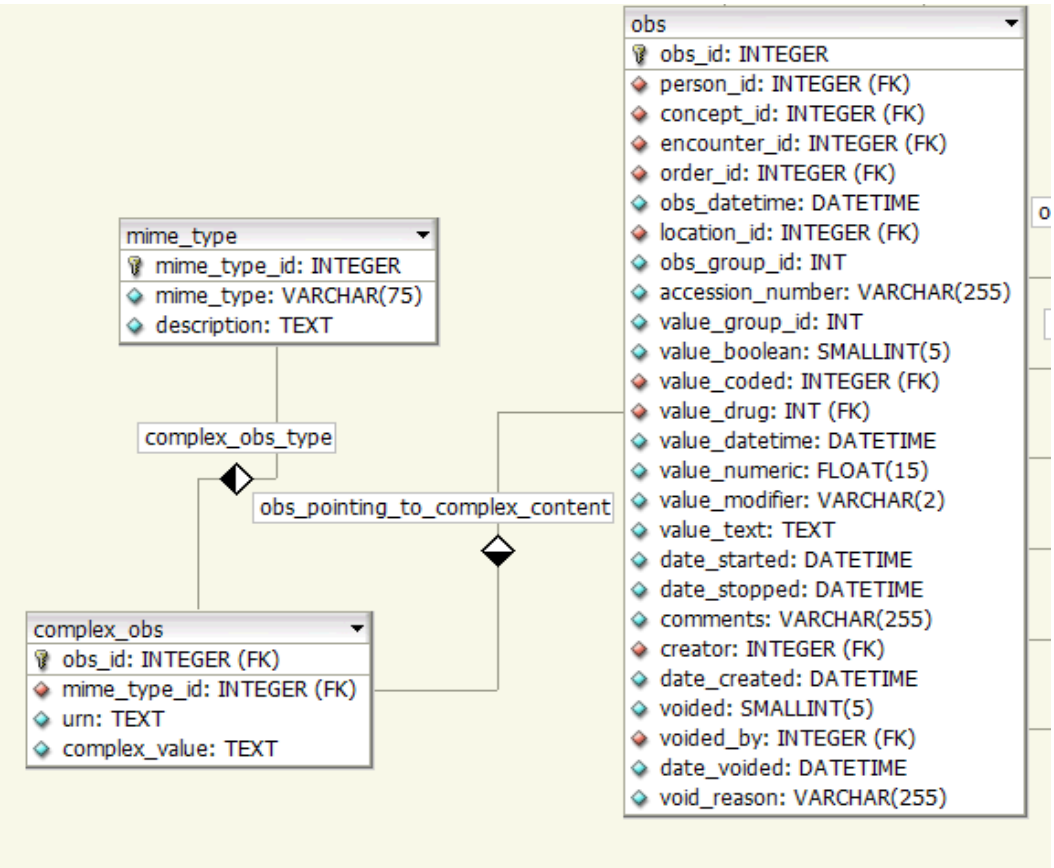


Figure: relation between mime_type, complex_obs, obs tables.

[source: http://openmrs.org/images/5/59/Openmrs_data_model_1.10.png OpenMRS Data Model access on 20rd april]

- Obs table: This table contains details regarding all the observation made during clinical setup.
- complex_obs: This table contains details regarding complex observations made during clinical setup such as X-Ray, Scan. It contains field “obs_id” which foreign key for a table which corresponds to record in “Obs” table. “mime_type_id” field indicate kind of file present at location mentioned by “urn” field.
- mime_type: This table contains detail regarding mime file types.

2.2 Module 2: Reporting ETL and OLAP

2.2.1 What is Data Warehouse ?

Data Warehouse is data storing repository where organizational data is stored which is used for reporting and managerial decision purpose and analysis of organizational data . Data Warehouse is made up of various components and layers which is explained in details as follows.

2.2.2 Components of Data Warehouse:

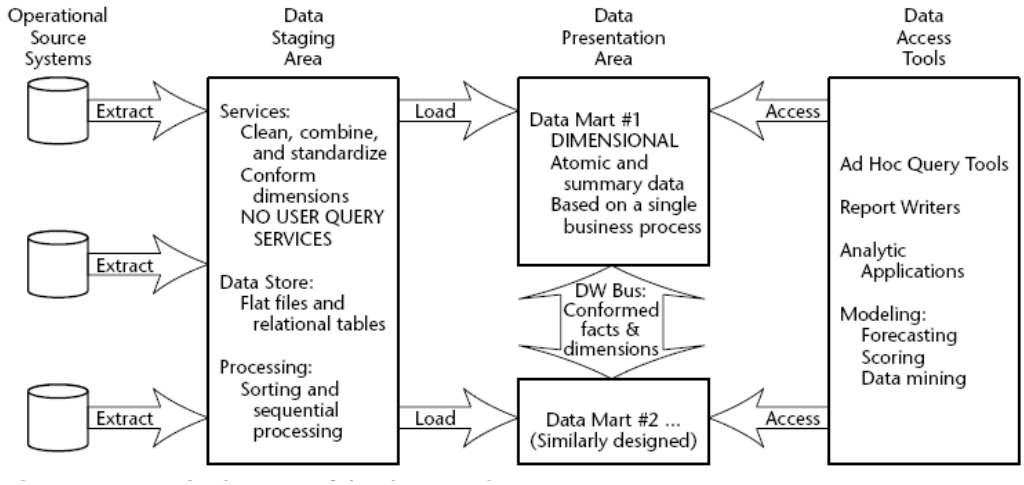


Figure 1: Components Of Data Warehouse.

[Source: Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 7]

Operational Source Systems:

In above diagram Operational Source system capture day to day transactions made by business some time it also referred as a raw data. We can consider that Operational source system is outside part of Data Warehouse we have very less control over the type and content of this data. As we can see from above figure all the data collected in Operational Source System is then feef to Data Staging Area as it is where actual ETL operations are performed.

Data Staging Area:
Data staging area or layer of Data Warehouse is area where ETL operations are performed. This stage is an interface between operational system and presentation layer. The main architectural concept of this area is business uses do not have any access to this stage and no query operations can be performed on this stage.

The first step of data staging area is extraction. In extraction we read and understand the source data from legacy system and then copy data needed for Data Warehouse. Once extraction of required data is done in first step then required transformations or business rules are applied on data. Transformation which are applied on data are like sorting data on a given key, changing dml of input file, grouping data for a given key, converting date formats, etc..This process is also known as cleansing data. All these transformations are need to be done before data can get loaded into data warehouse presentation area or database or files.

The data staging area is mainly consists of operations performed on data rather considering about relational view of data. Once we validate data for one to one or many to one fussiness rule there is no point to build third normal form physical database. Rather in many cases when data is extracted from operational source system they have already undergone the process of normalizing, in that case we only needed to clean data arrived at data staging area. without regarding whether data arriving at data staging area is normalized or not the final stage or step of staging area is loading data into Data warehouse. Different quality performance measure such as indexing than can be applied to data present in Data Warehouse.
The ETL architecture mentioned above can also be presented as follows :

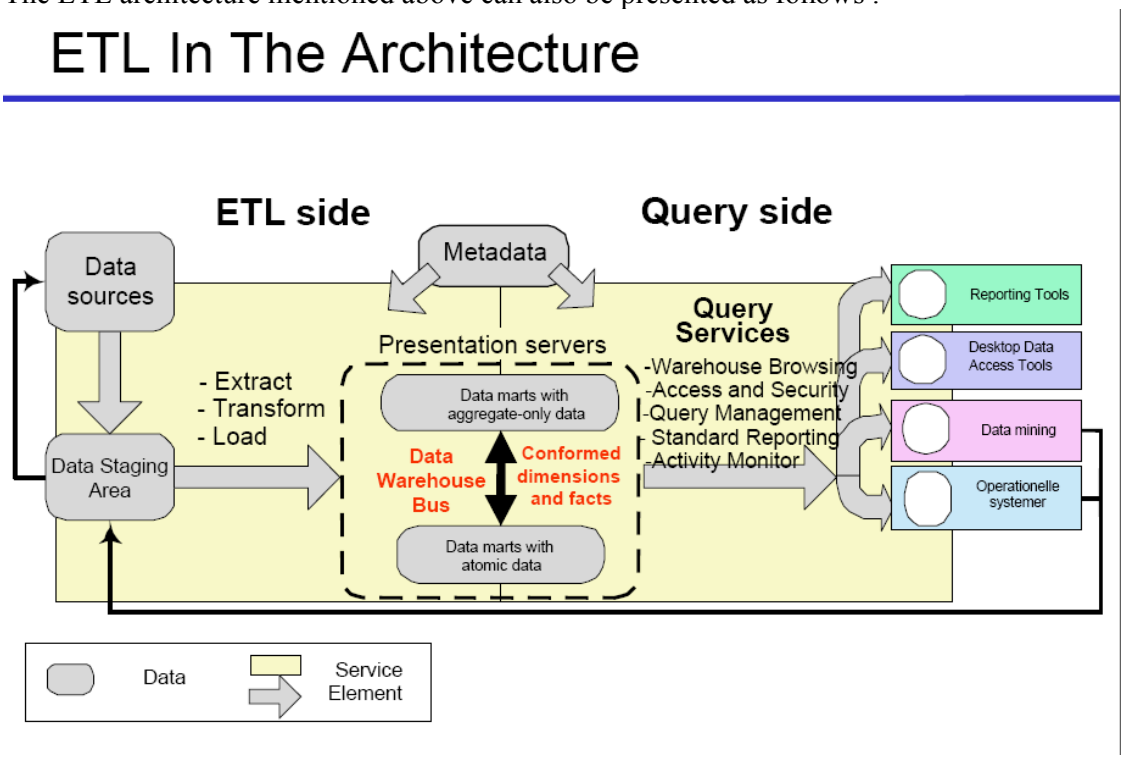


Figure: ETL architecture

[source: Torben Bach Pedersen Aalborg University ETL_Logic.pdf, page 7]

Data Presentation layer:
Data presentation area is the layer where data is stored by staging area and it is available for reporting teams, business users and all of those who want to access Data Warehouse. The data can be access by these people with the help of various data access to tools.

Presentation layer is made of series of data mart, where data mart is reforestation of data for a given single process or unit. As business have more than one unit or process it is oblivious there are number of data marts present in data warehouse. One of the important aspect of data mart is that, data represented in it must be detailed, atomic which is necessary to answer any ad hoc request access or queries made by users.

One of the required parameter of data warehouse is bus architecture. If data marts are built using dimensions and fact table then we can say that data warehouse conformed to bus architecture of data warehouse. In general total number of data marts present in any business organization depends upon how how much complexity of business want to aplly in data warehouse. Bus architecture is key to build a de centralized data warehouse.

"Data in the queryable presentation area of the data warehouse must be dimensional, must be atomic, and must adhere to the data warehouse bus architecture."

[citation : Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 13]

If presentation area of data warehouse is designed using relational database model, then they way these dimensional tables are modeled are called as **star schema.** This is because there will be one main fact table and many dimension table related with that fact table which look like star. Also if presentation area design on a basis of online analytic processing(OLAP) or multidimensional database then data is stored in **cubes.** In general data marts in data warehouse are updated quite frequently and it there is any data corrupted or missing then necessary changes are done to make sure data present in data mart is correct.

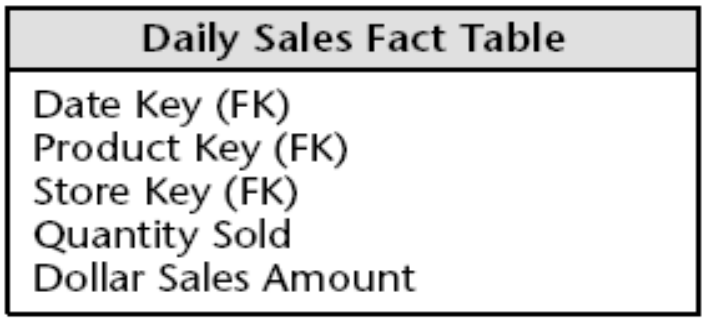
Data Access Tools:
Data access tool is last component in Data Warehouse environment. These are tools by means of which end users can apply queries to presentation layer where data is actually stored. Data access tools vary from the way user end result should be, as it can be simple ad hoc query or more sophisticated some data mining tools or modeling applications. In general in day to day business already built, parameter driven applications are used to query presentation layer. In some cases result are loaded back into wither of first three layer of Data Warehouse.

Star schema:
Before discussing what star schema is we will need to take look at fact tables and dimension tables to understand it.

Fact Table:
Fact tables are main or primary table which contains day to day business data. Fact tables generally contains primary keys of other tables which are referred here as a foreign keys.

Also along with foreign key fact table also contains few other attributes which are may be needed for a day to day business.

Following is example of simples fact table:



[Source: Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 17]
As shown in above figure the fields DateKey, ProductKey and StoreKey are foreign keys to Daily Sales fact table. Also this table contains some non key attributes such as Quantity Sold and Doller Sales Amount.

Dimension Table:
Dimension tables are relatively small tables which are used for look up purpose.

Figure: Simple Dimension Table

[Source: Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 20]

Total number of records in dimension table are less as compared with fact table but dimension table contains more columns or attribute. Each dimension table contains one primary key, as we can in above figure Product Dimension Table contain Product Key as a primary key. In some cases this key is just a numeric number incremented for each row and act as unique identifier for each row.

All the attribute present in dimension table are main source of query conditions, roll ups. Dimension table attributes play very important role in data warehouse as they are main source of all reporting functionality they plays key role in creating data warehouse which make data warehouse usable and understandable. To create a good data warehouse importance must be given to creating correct dimension tables.

"Dimension tables are the entry points into the fact table. Robust dimension attributes deliver robust analytic slicing and dicing capabilities. The dimensions implement the user interface to the data warehouse."

[citation : Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 20]

Dimension tables are generally denormalized tables. Total number of records present in dimension tables are also quite small. In general they constitute less than 10% of total Data Warehouse but they play vital role in functionality of data warehouse. As dimension tables constitute very small amount of data warehouse we always trade off database principles like normalizing or snow flaking for simplicity and accessibility.

Fact and Dimension tables together forms star schema:

Figure: Star Schema

[Source: Ralph Kimball and Margy Ross(2002) *The Data Warehouse Toolkit* second edition New York: John Wiley and Sons, Inc. Robert Ipsen page no 22]

As we can see above figure Daily Sales fact tables is related to all other dimension table. The above figure structure looks like star so we called such schema as star schema. We will discuss star schema in details and put forward why its important in design of data warehouse. Just by looking at above figure/schema we can notice one of the important aspect of this figure is simplicity understandability.

Following are the important things we can notice about star schema:

1. **Simplicity and understandability:** Just by looking at above figure/schema we can notice one of the important aspect of this figure is simplicity understandability. Any business user who can see this schema can easily under this and tell us whether this is their business or not.

2. **Performance:** Database administrator can easily make process this schema with less joins.

3. **Extendibility:** Extendibility is one another important use of star schema. Business need changes day by day. As we can see above, even if business changes their needs we can easily accommodate those changes in above by just adding new dimension tables and making reference to it in fact tables.

3 Action

3.1 Module 1 : Complex Obs Support
Software Specification:

Non-Functional Requirements:

Software
The system is to be written in the Java programming language with all code conforming to the standards. Also MySql database will be installed proper ally so that user program can communicate with this database in proper way.

Hardware
The system shall be suitable for deployment on computers operating within a large performance bracket. With performance in mind there are three key factors which need to be addressed, namely hard disk space, memory capacity and processing power. Processor power and memory capacity are not of key importance while running OpenMRS system.

Reliability
The final program shall be designed with reliability as a primary objective, and as such all th validation conditions are satisfied while implementing program. Furthermore, suitable error messages will be displayed in order to provide the most user friendly environment possible. Strict validation of human input will allow the system to run reliably and alleviate any errors that may be incurred through unexpected entry.

Security
Inbuilt security functionality of OpenMRS will be applied to the system.

Functional Requirements:
Project main functionality to implement is to extend Current API implementation of OpenMRS to deal with processing of the complex observations and loading data into these tables. By making use of the XML files or any other file formats it can be possible to deal with complex observations and loading data into database(complex_obs table), but system also needs to cope when total amount of data is huge or reference to another file system or database is given via URN or URL.

Error Handling:
System will try to make robust so that it can cope with any kind of errors that may occur.
Java exception handling will be implemented throughly to address this issue carefully.

3.2 Module 2 : Reporting ETL and OLAP

Software Specification
Non functional requirements:

Software:
Installing OpenMRS will be one of the basic task that needs to be done. Installing apache ant. Apache ant is software which is used for automatic software building process.

The functionality of Apache ant is same as make utility only the things is Apache ant is written in Java language and it is useful for projects written in java. Installing MySQL database. Installing Pentaho business intelligence server. Installing Kettle Data Integration tool.

Hardware
The system shall be suitable for deployment on computers operating within a large performance bracket. With performance in mind there are three key factors which need to be addressed, namely hard disk space, memory capacity and processing power.

Reliability
The final program shall be designed with reliability as a primary objective, and as such all th validation conditions are satisfied while implementing program. Furthermore, suitable error messages will be displayed in order to provide the most user friendly environment possible. Strict validation of human input will allow the system to run reliably and alleviate any errors that may be incurred through unexpected entry.

Security
Inbuilt security functionality of OpenMRS will be applied to the system.

Functional Requirements:

1. Design star schema
2. Implement star schema
3. Implement ETL functionality
4. Transfer data from OpenMRS to OLAP system by making use of ETL functionality
5. implement reporting functionality
6. Implement Data Warehouse / OLAP system.

Software Design:
3.3 Module 1: Complex Obs Support

Te which we needed to perform in this module is storing and retrieval of the images.
Storing image task:

- Write API to accept file from user.
- Once file accepted write method which connect to server where we suppose to store file.
- Write method which will query mime_type table to find mime_type_id for a given file type.
- Write method which will query complex_obs table to find urn to store file at a given specific location or database.
- Store the received file at location mentioned by urn.

Retrieving file:

- find obs_id of person.
- Write method to query complex_obs table to find corresponding mime_type_id and urn.
- Query mime_type table to find file type.
- Also write method to connect particular files system or server or database (if its not the same where all above tables present) and retrieve file.
- Write API to display retrieved image.

3.4 Module 2: Reporting ETL and OLAP

Software design

Designing of star schema

While designing star schema we need to understand the data model proposed by OpenMRS team. There are various factors which must taken into consideration while designing star schema. those factors as follows:

- Model star schema by making using of fact and dimensional tables concept. Dimensional tables are small, normalized lookup tables which contain details about very small entity where fact tables are very large table and they have many to one relationship with corresponding dimension table.
- We will try to avoid snowflake structure while designing star schema.
- Try to create dimensional hierarchies in correct way.

OpenMRS team has provided me with some sample report which we need to generate from the Data Warehouse we will be going to create.

Following is example

reports:

National Care and Treatm

			Cohort Jan 06	6 mo Jul 06	12 mo Jan 07	24 mo Jan 08	Cohort Feb 06	6 mo Aug 06	12 mo Feb 07	24 mo Feb 08
1	G	Started on ART in this clinic - original cohort	41	41	41		32	32	32	
2	TI	Transfers in	0	0	0		0	0	0	
3	TO	Transfers out	0	0	0		0	0	0	
4	N	Net current cohort	41	41	41		32	32	32	
5	H	On original 1st line regimen	27	18	14		17	16	0	
6	I	On alternate 1st line regimen (substituted)	0	1	1		0	2	0	
7	J	On 2nd line or other regimen (switched)	0	0	0		0	0	0	
8		Current regimen unrecorded	11	1	0		15	0	0	
9		Stopped / current ARV status unrecorded	3	3	1		0	2	2	
10		Died	0	0	0		0	0	2	
11		Lost to follow up / did not visit during month	0	18	25		0	12	28	
12		Number of cohort alive and on ART	38	20	15		32	18	0	
13		Percent of cohort alive and on ART	93%	49%	37%		100%	56%	0%	

NACP Cohort analysis Report

PEPFAR report

01-Oct-06 to 31-Dec-06

Section	**1**	**HIV Palliative Care (non-ART and ART care)**			
Item	1.1	Cumulative number enrolled in HIV care by the beginning of period	4,387		
				Male 0 - 1	18
				Male 2 - 4	52
				Male 5 - 14	228
				Male 15 and over	825
				Female 0 - 1	29
				Female 2 - 4	68
				Female 5 - 14	263
				Female 15 and over	2,904

PEPFAR report

To design star schema we need understand first Data model proposed by OpenMRS team. Following tables are present in data model presented by OpenMRS team.

patient_identifier	note	concept_name	user_role
cohort_member	concept_map	concept_source	privilage
cohort	concept_proposal	drug_ingredient	role_privilage
location	concept_datatype	drug	notification_alert
mime_type	concept_answer	concept_set	field
complex_obs	concept_synonym	concept_word	field_type
obs	program	patient_state	form
encounter	concept_state_conversion	person_attribute	form_field
patient_program	person	person_attribute_type	field_answer
patient_identifier_type	person_name	user_property	orders
patient	program_workflow_state	users	drug_order
relationship_type	concept_class	role_role	order_type
relationship	concept_numeric	role	scheduler_task_config_property
program_workflow	concept	notification_alert_receipient	global_property
encounter_type	scheduler_task_config		

Table : Tables in Data Model proposed by OpenMRS team

After going through data model and report we can found the tables which we will may be needed needed to built star schema and need to look properly are:

person
cohort_member
cohort
patient_state
person_attribute
person_attribute_type

Implement star Schema:
Once star schema is ready install MySQL and implement star schema in MySQL. While implementing star schema make sure it is implemented as renormalized SQL schema.

Design ETL functionality:
To implement ETL functionality we need to have KETTLE of Pentahos BI server. Try to workout with Pentaho's Ketle spoon to determine what kind of ETL functionality we can implement. Implement ETL scripts to move data from OpenMRS to Data Warehouse.

Implementation of Data Warehouse:
Designing of cubes and for reporting and analysis purpose.

3.5 Plan:

I have selected two modules of OpenMRS to implement and as ETL/OLAP module is big as compared with Complex Obs Support i am planning give more time to ETL/OLAP module. While working on ETL/OLAP module i will implement prototype for Complex Obs Support. Following is Plan of working on both modules in terms of week.

Week 1: (June 8- 14)
ETL:
Install and deploy the OpenMRS system. Work with OpenMRS mentor to finalize requirements and scope of project. Try to find general patter of reporting questions. Work out with preliminary design with Star Schema.
Complex Obs Support:
Work with OpenMRS code base to understand which modules will be needed to interact with complex observation functionality.
Report:
Include work done in this week in dissertation report.

Week 2: (June 15- 21)
ETL:
Analyze the reports given by OpenMRS team. Design Star Schema as a demoralized SQL schema. Work on star schema design.

Complex Obs Support:
Implement upload image prototype functionality to store complex observation into system.
Report writing:
Include work done in this week in dissertation report.

Week 3: (June 22- 28)
ETL:
Work on design of star schema. Start working on implementing star schema in MySQL.
Complex Obs Support:
Implement/Test upload image functionality to store complex observation into system.
Report writing:
Include work done in this week in dissertation report.

Week 4: (June 29- July 5)
ETL:
Design of star schema
Complex Obs Support:
Implement/Test upload image functionality to store complex observation into system.
Report writing:
Include work done in this week in dissertation report.

Week 5: (July 6 - 12)
ETL:
Implement/Test Star Schema in MySQL.
Report writing:
Include work done in this week in dissertation report.

Week 6: (July 13 - 19)
Buffer time:
Finish with all the tasks scheduled before that week

Week 7: (July 20 - 26)
ETL:
Work with Pentaho's KETTLE to find out ETL functionality available.
Complex Obs Support:
Write prototype to retrieve image functionality. Also start coding for retrieve image functionality.
Report writing:
Include work done in this week in dissertation report.

Week 8: (July 27- Aug 2)
ETL:
Implement/Test ETL functionality of OpenMRS using Pentaho's Kettle.
Complex Obs Support:
Implement/Test retrieves image functionality to retrieve complex images from database or file system.
Report writing:
Include work done in this week in dissertation report.

Week 9: (Aug 3 – Aug 9)
ETL:
Write/Test script to load data from OpenMRS to Datta Warehouse or OLAP.
Complex Obs Support:
Black box testing for retrieve image and upload image functionality.
Report writing:
Include work done in this week in dissertation report.

Week 10: (Aug 10- Aug 16)
ETL:
Write/Test script to load data from OpenMRS to Datta Warehouse or OLAP.
And Implement new interfaces which will use the above mentioned ETL functionality.
Report writing:
Include work done in this week in dissertation report.

Week 11: (Aug 17 – Aug 23)
ETL:
Write script to load data from OpenMRS to Datta Warehouse or OLAP.
Implement new interfaces which will use the above mentioned ETL functionality.
Start with Design cubes based on reporting and analysis needs.
Report writing:
Include work done in this week in dissertation report.

Week 12: (Aug 24 – Aug 30)
Buffer time:
Finish with all the tasks scheduled before that week

Week 13: (Aug 31 – Sept 6)
ETL:
Design cubes based on reporting and analysis needs.
Evaluate work done on Complex Obs Module

Week 14: (Sept 7- Sept 13)
Evaluate work done on ETL ModuleReport Writing:
Report writing:
Writing of Dissertation.

Week 15: (Sept 14 – Sept 21)
Writing of Dissertation.

Duration (Sept 22 – Sept 30)
Writing of Dissertation.
Submission of Dissertation.

Gant chart

Preliminary design with Star Schema. (ETL)	X					B						B			
Work with OpenMRS code base (Complex Obs)	X					U						U			
Design Star Schema (ETL)		X	X	X		F						F			
Implement upload image prototype (Complex Obs)		X				F						F			
Implement/Test upload functionality(Complex Obs)			X	X		E						E			
Implement/Test Star Schema in MySQL(ETL)				X	X	R*						R*			
Write prototype to retrieve image(Complex Obs)							X								
Implement/Test retrieve functionality(Complex Obs)							X	X							
Implement/Test ETL functionality(ETL)								X	X						
Write/Test ETL script to load data (ETL)									X	X	X				
Black Box and White Box testing for Complex Obs									X						
Implement new interfaces for ETL (ETL)										X	X				
Design cubes based on reporting (ETL)											X		X		
Evaluate work done on Complex Obs Module													X		
Evaluate work done on ETL Module														X	
Dissertation write-up	X	X	X	X	X		X	X	X	X	X			X	X	X
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
* Buffer time is added to finish with all the taskes scheduled before that week	8th june	15th june	22nd june	29th june	6th july	13th july	20th july	27th july	3rd aug	10th aug	17th aug	24th aug	31st aug	7th sept	14th sept	21st sept

4 Risk Assessment :

1) Unclear requirement specification:

Impact:
Understanding all requirements specification and analysis of these specification is important for any software project. Before design and implementation of software user must understand what are the needs and requirements of these software. Any misunderstanding of these requirements may lead to complete failure of software implementation.

Probability:
If user of system and implementer of system have not agree of final system requirements of system or if there is miscommunication between two while understanding of these requirements, above risk may occur.

Rank:
1st

Management:
In order to successful implement the software project user and implement must agree on system requirements. Scope of system must be defined and both sides should agree on it. Also review of requirements specification must be done before starting with design phase of system.

2) Communication

Impact:
As OpenMRS is system developed by many volunteers across the world, some time getting help on the software specification or requirement details may take longer time than expected. In such case development of any module will take longer time than expected as what exactly needed to do not much clear to developer of module.

Probability:
Some time communication via email or other communication facilities takes longer time than expected and developer may get stuck while implementing some functionality. To resolve these problem he/she may need to contact project mentor and explained him the problem. Then mentor will think on it and suggest other solutions or overall requirement may be needed to change . All process will take longer time than expect which caused unnecessary delay in implementation of software.

Rank:
2nd

Management:
A common and efficient way of communication must be decided to avoid communication gap and proper agenda must be decided to resolve any problem which may occure while implementing project.

Reference:

[1] Ralph Kimball and Margy Ross *The Data Warehouse Toolkit* second edition pages 1-17, New York, USA, 2002, John Wiley and Sons.

[2] Bert Scalzo *Oracle DBA Guide to Data Warehousing and Star Schemas* Chapter 4. Star Schema Universe, New Jersey, USA, 2003, Pearson Education

[3] Oracle *Oracle Database Data Warehousing Guide, 10g Release 1 (10.1)* CA,USA 2001, 2003 Oracle Corporation

[4] Oracle *Oracle9i Data Warehousing Guide Release 2 (9.2)* CA,USA*)* March 2002Oracle Corporation

[5] Muhammad Ahmad Shahzad*, Data Warehousing with oracle* Oracular, Inc 317

[6] OpenMRS An EMR for Developing Countries Burke Mamlin, MD Paul Biondich, MD MS …and the OpenMRS collaborative page page no 24

[7] Torben Bach Pedersen Aalborg University ETL_Logic.pdf, page 7

[8] http://openmrs.org/wiki/Reporting_ETL_and_OLAP_Project

[9] http://openmrs.org/wiki/OpenMRS_Overview

[10] http://openmrs.org/wiki/Obs_Table_Primer

[11] http://openmrs.org/images/5/59/Openmrs_data_model_1.10.png

[12] http://openmrs.org/wiki/OpenMRS

[13] http://openmrs.org/wiki/Projects#Complex_Obs_Support

Abbreviations:

[1] ETL: Extraction, transformation and Load.

[2] OLAP: Online Analytical Processing

[3] OpenMRS: Open Medical Record system