The Data Driven Hospital

Data Engineering Audit and Recommended Approach

Gurjus Singh

Northwestern University

SECTION I

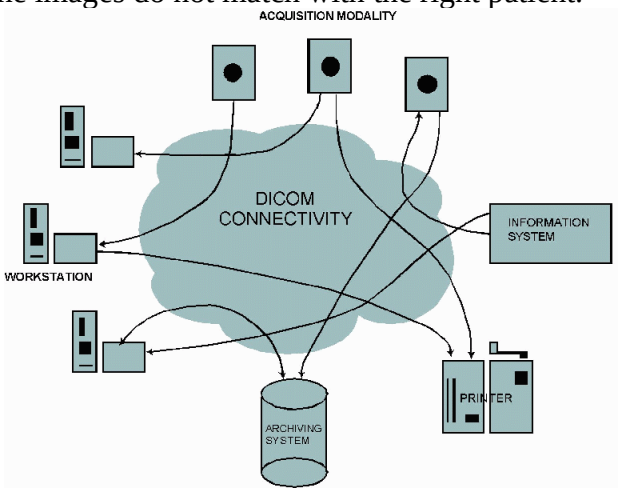
Introduction

I have currently been working at Oroville Hospital for about two years. From examining all the different applications we use such as Health Record Applications, Dicom Systems, and Excel sheets and taking classes in the Northwestern University program, I think this can be a good opportunity for Oroville Hospital to become more technologically savvy in the future. Firstly, I believe we are just at the beginning in terms of becoming "analytically competitive" (Davenport, 2017, n.pag). This is a term that Davenport (2017) uses in his book for companies that have a niche in terms of where they use their analytical tools.

For us to become analytically competitive our hospital has to become more efficient. This can be achieved by using more numbers, gathering more data, and hiring more data analysts, as well as being able to train the employees we already have to have more of a technical skill level. There are some advantages to the way we are using image processing systems, but it would be nice to know the backend details of how the data is being processed. We are also at a disadvantage relying heavily on Excel sheets, which are highly inefficient to track errors, patient data, and load data. I will describe what we should keep in place that I think is important for the hospital and also suggest alternatives to what I think we can improve on.

Current system overview

One thing I like that the hospital is using already is the DICOM system, Digital Imaging and Communication in Medicine ("DICOM", para 1). I think it is important that we keep this system in place because DICOM is a way to transmit images to a patient record and in this case through our Electronic Health Record Application Tenzing CPRS. I know many different machines such as scanners, servers, Picture Archival and Communication Systems (PACS), workstations, printers, and network hardware which are used to transmit these images and that DICOM is an international standard for many hospitals which is why I suggest this an advantageous system to keep in place ("DICOM", para 2). I also realized that TCP/IP are the primary ways for which these images get across ("DICOM", para 2). I am responsible for helping with monitoring DICOM to make sure machine images are flowing across and helping correct DICOMs when the images do not match with the right patient.



Here is a diagram of how DICOM works ("NEMA", n.page)

Another interesting type of tool I have learned about while at the Hospital is the Mirth Interface. HL7 messages are important to the Mirth Interface application because it allows doctors to create notes and send them to Tenzing Applications which are responsible for patients records. For example, when a doctor dictates a study, they send it as a message in Mirth. When researching about Mirth I saw it is a HL7 message interface ("Why Choose Mirth" para. 4). They describe it as a middleware which is mentioned in the Data Intensive Applications book as a type of "web service" that allows one service to make requests to another service within an organization (Kleppmann, 2017, pg 132). Mirth is a type of web service that allows the transformation, routing and filtering of data in the organization ("Why Choose Mirth" para. 6). It supports several protocols such as HTTPS and SOAP to name a few ("Why Choose Mirth" para. 8). During my time at Oroville Hospital, my experience with the Mirth interface has been assigned to fix HL7 messages so they show up correctly in Tenzing, which is an application which shows patient vitals and records when they are admitted to the hospital. This has allowed me to learn more about this interface and how it is important to data in the hospital setting.

Thirdly, we have to keep track of the database statistics. This includes, load and reference counts, which is the number of references to data in a database, and also noting pharmacy medicine prescription errors in Excel sheets. I have learned though the Data Science program that Excel sheets are not the best way to keep track of data and we need to think of another solution to these tasks ("Reference Count", n.pg.). Another task we do using Excel sheets is keep track of duplicate patient profiles that need to be merged together. Instead of using Excel sheets, I suggest an approach suggested below.

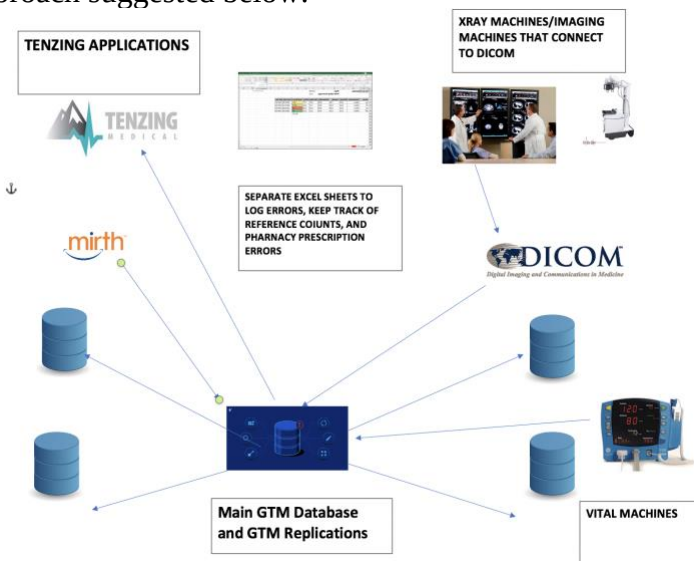


Figure 1-1 Current Oroville Hospital System

In terms of our current system at Oroville Hospital, Figure 1-1 shows our current system and the relationships involved. We see there is a main GTM database. A GTM database is known as a key-value database engine, which involves a high throughput ("GT.M" para 1). According to Kleppman (2017), high throughput means that this type of database can process a high number of records per second (pg. 13). The primary language used for this type of database is called MUMPS which is specifically designed for the healthcare industry ("MUMPS" para 1). I know

that we do use replication databases to keep our data systems safe in the event of emergencies. This is the main purpose of using replication as described in the course I took.

Lastly, the figure above shows the applications the database is connected to such as the Tenzing Applications which is used by our main users, such as the doctors, nurses and other healthcare providers. Mirth is used more for the backend and makes use of batch processing. It also makes sure data gets to the database. There is also the element of the vital machines which collect data on patients in the hospital.

PART II

Recommended Approach

As mentioned above, there are a couple of problems with the current approach on how our data resides. For example, as a data scientist I am confused on why some of our data is inputted into Excel sheets which is a very inefficient way of holding data and the reason is that it becomes very slow to access Excel files as more people use it and allows only one person to manipulate the file. In a sense a database is much faster and allows several users to manipulate the database at once. Another reason why Excel is inefficient is in terms of views. With a database, users can customize their views to see their own view of the data, with an excel sheet this cannot be done. Thirdly, when handling big amounts of data, Excel only has a limited amount of data it can store. For this big data, I believe AWS is the way to go.

For this type of data, I suggest we set up another database which will save all the information that a user can inputs. In our class, we learned about Event Logs and how they are responsible for keeping track of events that interact with the database ("Event", n.pg.) . I believe we can create something along these lines at the hospital. Event Logging can be useful for database monitoring and a way of reporting problems with our database such as when the system load is high and when we need to keep track of the reference count.

We can use a hybrid model, where we keep patient data on the premise database system for security purposes and use a cloud database to keep track of the data about errors. The type of information we can store is time error occurred. For example, when we receive errors regarding prescriptions to be sent to the pharmacy, we can log these errors in the database. We can create a dashboard to track these errors. The dashboard can be connected to the cloud database and the way it will work is the healthcare user submits information regarding the errors such as time, type of error: (database error, medication error, duplicate patient error) and source. This will significantly reduce the amount of Excel sheets we use at our workplace. The application which the end user will use will be a web-based form, where it will have fields to submit the information on the error seen. The storing of data can be based on a key-value pair, where the key is the time and date of the error as suggested in Data Intensive Applications book (Kleppman, 2017, pg. 202). In this way partitioning of the database will be easier, and data will be faster to find.

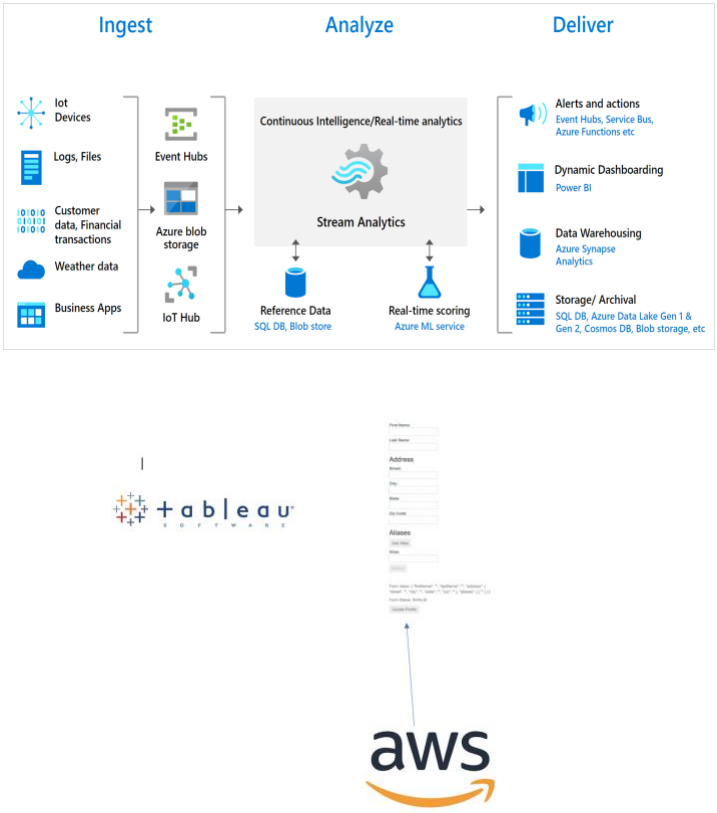
As I suggested previously, we should use a cloud database for these types or errors as this data is less sensitive if stolen. It will also lower the cost in comparison to extra resources when building an on-premise database like we currently have. I believe that Amazon Web Services is my recommended cloud database as it is flexible. It can allow the storage of non-relational data, and we can also store relational data if needed. We can also access this database remotely, and also speed is better. You also can get as many resources as you need and not more as compared to an on-premise database ("AWS", n.pag.)

I do believe there are some flaws when using a cloud database storage. Security is the main concern. There is always the risk of hacks; although this is a main concern; we can rest

easy knowing that Amazon provides TLS encryption and enterprise level security (Underwood, 2020, para 7). In regards with HIPAA, Amazon has worked with the top government officials in order to comply with this policy (Underwood, 2020, para 8). Another big flaw when thinking about the cloud is moments when your data is inaccessible. I think this is huge when the hospital will have projects which need to be done in a timely manner. With this said if the hospital does not have big concern over the data it has; I believe the cloud is a good resource we should invest in.

Another recommendation for Oroville Hospital is to consider adding stream analytics. I think this will be very beneficial as we already collect patient vitals in real time. We can add the Azure Stream Analytics service to our existing database to analyze the data. With this type of system in place we can find out how many patients are in the hospital at a given time, calculate the mean of the population and other statistics in near real time. I think we already have a hybrid model as it happens every 4 hours, but a streaming system will allow us to make real time decisions in emergency situations ("Azure", n.pag.).

Figure 1-2 with proposed elements



As explained, Figure 1-2 has recommended tools to incorporate in our organization. We should use a hybrid model using on-premise databases for streaming analytics as the diagram above shows how this will work. We will get our data from vital machines, then we can analyze this data in real time, and deliver the data to our CEO. I also suggest using a cloud database as it is less costly for storing error information instead of using Excel sheets. Excel sheets should be more for analyzing data, and not storing this information. I suggest we make a web form to allow

users to submit errors they see. I also show Tableau as this should be a front-end application to analyze batch processing data that we already have.

In order to use the hybrid model, I think the agile methodology is the best way to deploy the cloud database. The reason why this is the best model is because it is an iterative methodology which allows the developers to test and evaluate. ("Agile", n.pag.). This will allow developers to make and fix their mistakes depending on feedback given ("Agile", n.pag.). In regard to transitioning their data on the cloud, AWS also gives organizations an "ultra-capacity" external hard drive to make their transition smoother (Underwood, 2020, para 8). The hard drive is called AWS Snowball (Underwood, 2020, para 8).

Conclusion

Overall, I think Oroville Hospital needs to first start by creating a Data Science department to handle the big data. Once we have made our department, then we can start implementing this recommended approach. I do see several things already incorporated from taking the Data Engineering course, but I still think that our system is outdated. I see other students using advanced technologies such as Python and R, and we are stuck with SQL for right now.

References

Azure stream analytics. (2020). Microsoft Azure. https://azure.microsoft.com/en-us/services/stream-analytics/

Davenport, T. H., Harris, J., & Abney, D. (2017). *Competing on analytics: The new science of winning; with a new introduction* (Revised ed.). Harvard Business Review Press.

Kleppmann, M. (2017). *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems* (1st ed.). O'Reilly Media.

NEMA diagnostic imaging and therapy Systems division. (2020). NEMA.

http://dicom.nema.org/dicom/geninfo/brochure/BROCH96.HTM

Six advantages of cloud computing - overview of Amazon Web Services. (2020). AWS. https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html

Technosoft, T. T. (2019, November 11). *Why choose Mirth connect as your HL7 interface engine?* Technosoft Solutions. https://techno-soft.com/why-choose-mirth-connect-as-your-hl7-interface-engine.html/

Techopedia.Com. *Event log*. (2020). https://www.techopedia.com/definition/25410/event-log-networking

Underwood, N. (2020, May 25). *Is AWS More Secure Than On-Premise?* Privo.

https://www.privoit.com/resources/is-aws-more-secure-than-on-premise

Wikipedia contributors. (2020b, August 26). *Agile software development*. Wikipedia. https://en.wikipedia.org/wiki/Agile_software_development

Wikipedia contributors. (2020, August 8). *DICOM*. Wikipedia. https://en.wikipedia.org/wiki/DICOM

Wikipedia. *Reference counting*. (n.d.). https://en.wikipedia.org/wiki/Reference_counting