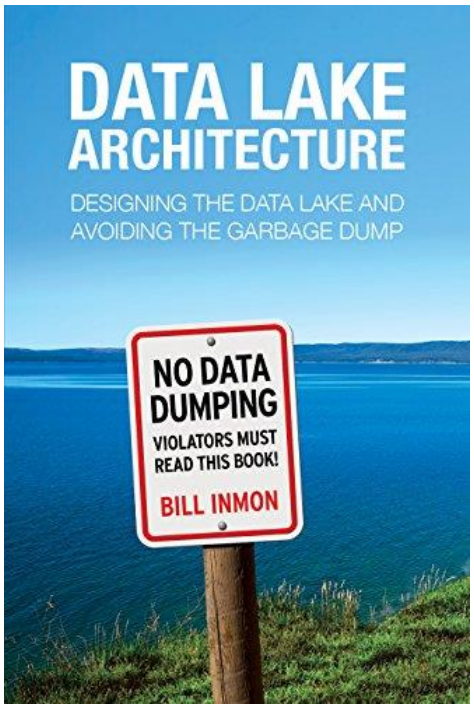
DATA LAKE ARCHITECTURE BY BILL INMON

A SUMMARY BY DR. ALVIN ANG



CONTENTS

DATA LAKE

WHAT IS A DATA LAKE

- Organizations store their Big Data in structures called "Data Lakes." (Inmon 2016)
- Often, only large corporations have data lakes.
- A data lake is a system or repository of data stored in its natural/raw format, usually object blobs or files. (Wikipedia 2020)
- A data lake is usually a single store of all enterprise data including raw copies of source system data and transformed data used for tasks such as
    - Reporting,
    - Visualization,
    - Advanced analytics and
    - Machine learning. (Wikipedia 2020)
- A data lake can include
    - Structured data from relational databases (rows and columns),
    - Semi-structured data (CSV, logs, XML, JSON),
    - Unstructured data (emails, documents, PDFs) and
    - Binary data (images, audio, video). (Wikipedia 2020)
- A data swamp is a deteriorated and unmanaged data lake that is either inaccessible to its intended users or is providing little value. (Wikipedia 2020)
- Big Data technology was best typified by the Hadoop Distributed File System (HDFS). (Inmon 2016)
- HDFS is an open-source software framework designed to store and process massive datasets distributed among many different computer clusters.

PROBLEMS WITH DATA LAKE

ONE WAY DATA LAKE

- ***One Way Data Lake*** = Data Dumping into the Data Lake without any analysis taken out.
- As data in the data lake continued to grow, organizations could do little of value with their treasure.

DATA SCIENTISTS

- ***Data Scientists*** = Specialists who make sense out of data lakes.
- Data Scientists are:
    - Hard to find
    - Expensive to hire
    - Hard to get their time when they are hired.
- No matter how well organized, when the data lake can be operated only by a few people (Data Scientists) whose cost is high and time is precious, the data lake just has limited corporate value.
- Despite the costs sunk into research, Big Data was just as brand new and unexplored for the scientists as the organizations. (Inmon (2016)'s book was written in 2016)

MOUNTAINS OF IRRELEVANT INFORMATION

- While collecting the data was a piece of cake, plucking something useful from this sea of knowledge was the real challenge.
- The analyst is buried behind mountains of irrelevant information.
- Given the sheer volume of data found in the data lake, the blandness of useful data makes it that much more difficult to find.

INACCESSIBLE METADATA

- Metadata is the description of the data (as opposed to the raw data).
- Metadata is used by the analyst to decipher the raw data found in the data lake.
- Example, if tracking visits, clicks and engagement to a website, metadata would include the IP address/geographic location of the visiting computer.
- The analyst never knows the meaning or source of the data that has found its way into the data lake. (What is the context of the data?)

LOST DATA RELATIONSHIPS

- The pool is so large that important data relationships are not carried forward into the data lake.
- It's considered too cumbersome to carry data relationships into the data lake.

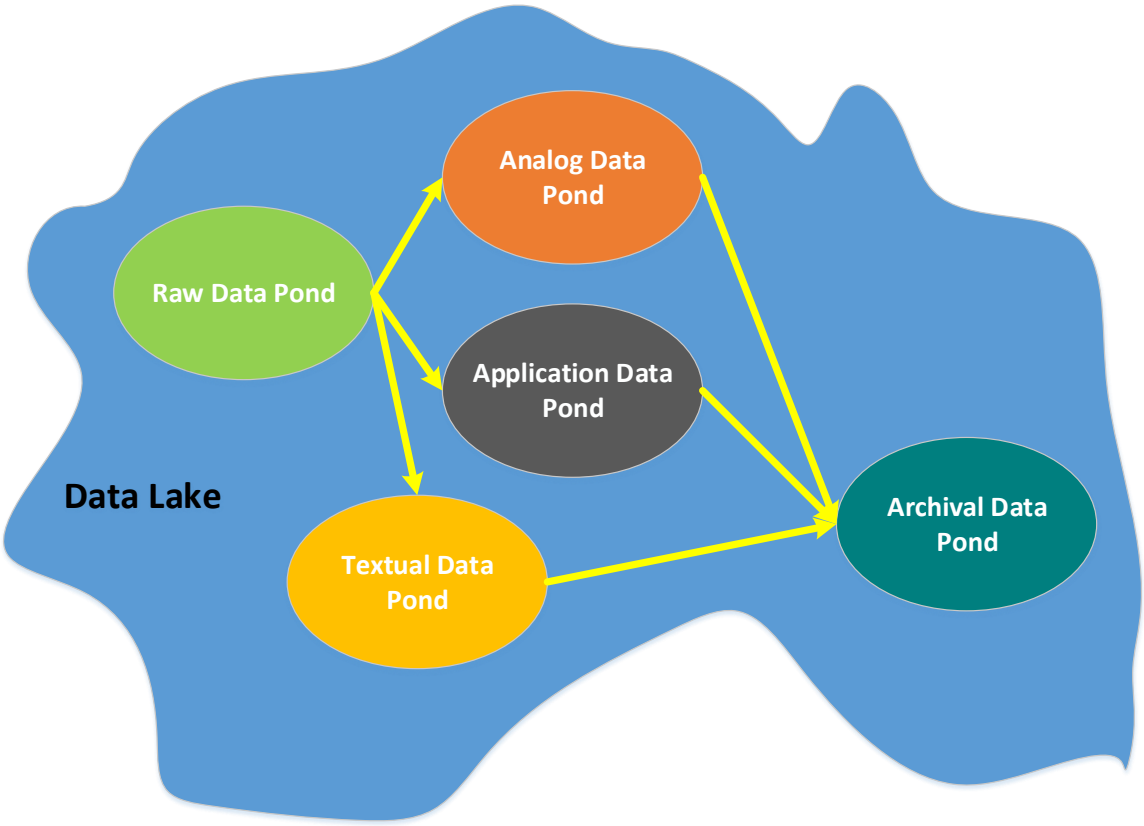


Figure 1: Various Ponds within the Data Lake

- Figure 1 shows the various ponds within the Data Lake.
- The genesis / beginning of Data Lake is the raw data pond.
- Raw Data Pond is a "holding cell" for raw data.
- Once raw data passes out from the Raw Data Pond, it is removed.
- Because the raw data has already served its purpose.
- The passing of data from Raw Data Pond to other ponds should be as quick as possible.
- ***Conditioning*** of data happens as raw data passes thru the various ponds….(that is, Raw Data Pond → Analog / Application / Textual Data Pond → Archival Data Pond)

- ***Conditioning*** = Preparing the data for analysis

- Can analytical processing be done for the raw data in the Raw Data Pond?

- Yes… but it requires a ***data scientist*** to do the analysis.

- Better analysis can be done only after it has been conditioned.

- Once the data has been conditioned, it can then be analyzed by the ***ordinary business user*** (at the Archival Data Pond).

- Analog / Application / Textual / Archival Data Ponds will be described in the later chapters.

- The purpose of the ***Archival Data Pond*** is to hold data that is not actively needed for analysis but might be needed at some future point in time for analysis.

POND DESCRIPTOR

- Pond Descriptor = A description of where the data in the pond originated from.

UPDATE FREQUENCY

- Update Frequency = the frequency of refreshment cycle which data is sent to the pond

SOURCE DESCRIPTION

- In many cases, data will pass through more than one source.

- Source Description = describes the lineage of the data (where it comes from).

VOLUME OF DATA

- Volume of Data = how much data is in the data pond. (measured in terms of number of records and in number of bytes)

SELECTION CRITERIA

- Selection Criteria = criteria used to select data for inclusion in the data pond.

SUMMARIZATION CRITERIA

- Summarization Criteria = a description of the algorithmic processing used in the shaping of the data in the data pond.

ORGANIZATION CRITERIA

- Organization Criteria = how the pond is organized.
- The organization of data can be rigorous or casual.

DATA RELATIONSHIPS

- Data Relationships = Relationships among the data found in the pond.

POND TARGET

- Pond Target = A description of the relationship between the business of the corporation and the data inside the pond.
- Typical pond target elements include:
    - customer profile,
    - sales record,
    - shipment record,
    - patient record,
    - part number,
    - inventory,
    - SKU,

- telephone call record,
- click stream activity,
- delivery information,
- insurance claim,
- flight schedule,
- passenger record

POND DATA

- Pond Data = physical data that resides inside the pond.
- In the world of Big Data, it is customary for the information to be stored in a "schema on read" manner.
- In this system, the data is initially stored in a block of data.
- Then when a query is made against the data, the system goes and reads the block of data and determines the schema inside the block.
- By organizing data in this manner, very large amounts of data can be stored efficiently.
- However, by storing the data in a "schema on read" manner, the retrieval and analysis of the data can cause significant overhead for the system to bear.

POND METADATA

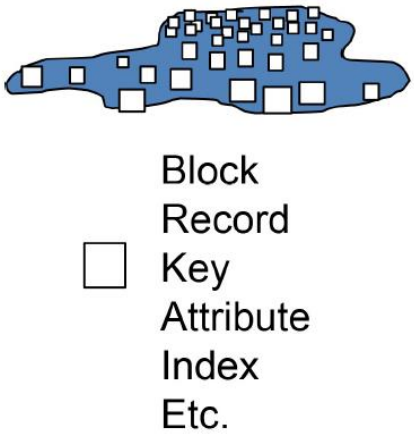


Figure 2: Pond Metadata (Inmon 2016)

- Figure 2 shows the types of Metadata stored in a pond.
- Metadata =
    - Records,
    - Attributes,
    - Keys, and
    - Indexes
- Pond Metadata = Describes the physical characteristics of the data contained in the data pond.
- If the data is stored outside the pond (in a standard Data Base Management System (DBMS)), the analyst can expect to find the same Metadata (records, attributes, keys, and indexes) carried inside.
- But if the data is stored outside the pond (but in document form), then the analyst can expect to find the data organized in a document by document organization. (Metadata not carried inside).

POND METAPROCESS

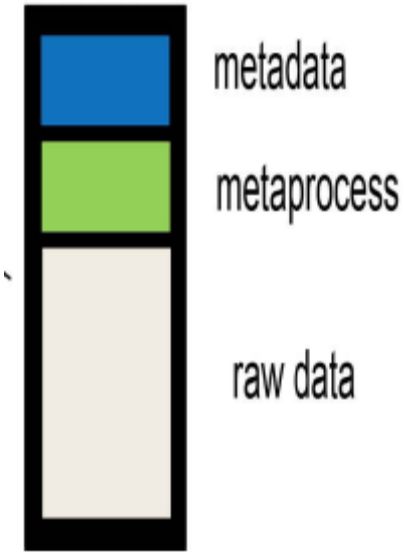


Figure 3: Difference between Metadata vs Metaprocess (Inmon 2016)

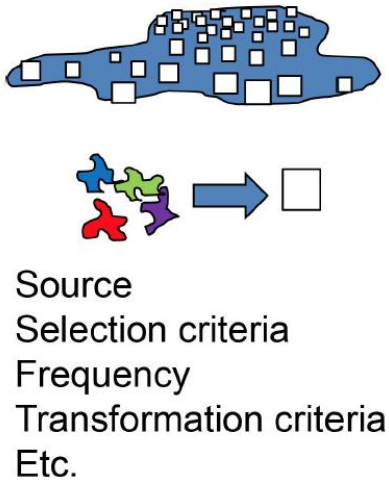


Figure 4: Meta Processing (Inmon 2016)

- Metaprocessing = transforming the data inside the data pond (to make it useful).
- Examples of Metaprocess Information:
    - When was the data generated?

- Where was the data generated?
- How much data was generated?
- Who generated the data?
- How was the data selected to be placed in the data lake?
- Once inside the data lake, was the data further processed?

- These are useful to the analyst.

POND TRANSFORMATION CRITERIA

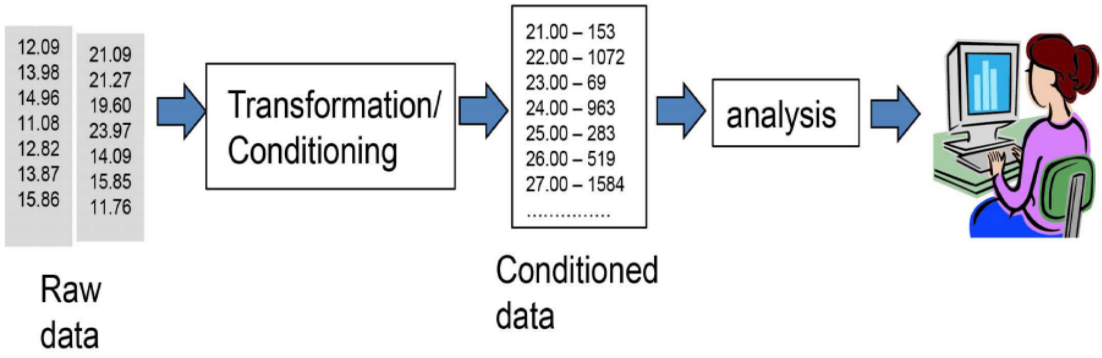


Figure 5: Transformation / Conditioning (Inmon 2016)

- Pond Transformation Criteria = documentation / criteria of how the transformation / conditioning of data inside the pond should occur.

- Transformation Criteria Example for **Analog Data Pond**:
  - "If length > 45 cm
    - then capture the record,
    - else do not capture the record."
  - "Catch all measurements of a certain machine for the month of May."

- Transformation Criteria Example for **Application Data Pond**
  - "If gender = 0 then convert gender to female.

- If gender = 1 then convert gender to male.
- If gender = x then convert gender to female.
- If gender = y then convert gender to male, and so forth."
- "If measurement is made in inches, then convert to centimeters."

- Transformation Criteria Example for **Textual Data Pond**
  - "If word = Honda then add car to classification.
  - If word = Porsche then add car to classification.
  - If word = Ford then add car to classification.
  - If word = Volkswagen, then add car to classification."
  - "If word = elm then type = tree.
  - If word = oleander, then type = bush."

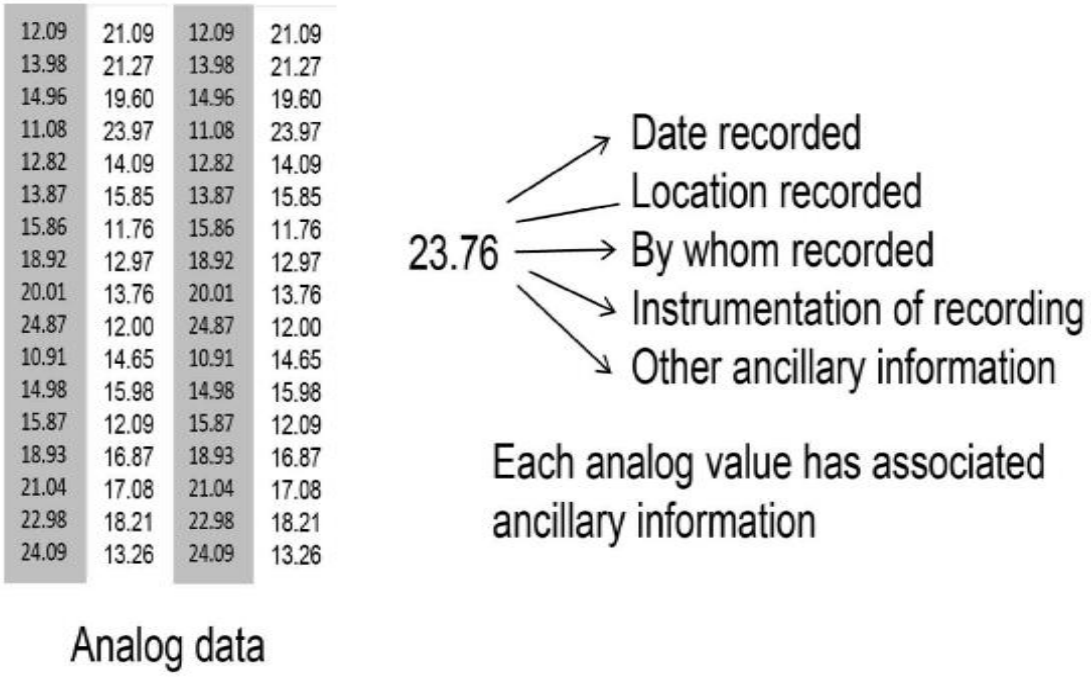


Figure 6: Analog Data (Inmon 2016)

- Analog data is generated by a machine, very voluminous and very repetitive.
- Analog data is made mechanically, without any user input or extra processing.

EXAMPLES

- Heat
- Weight
- Chemical composition
- Size

USAGE

- Analog data is used to signal to the analyst the cause of variation in measurement
- E.g. Machine Misalignment. (Check for outliers... what caused the misalignment…)
- Metaprocess Information associated with Analog Data is more important than the data itself.
- Examples of Metaprocess Information:
    - Time of measurement
    - Location of measurement
    - Speed of measurement

TRIGGERED BY…

- Analog Metaprocess Information is triggered by a manufacturing event, example:
    - A part is created
    - A shipment has been sent.
    - A box has been moved.

STORAGE

- Analog measurements are stored in log tapes.
- Numbers generated in very small intervals.
- The format of the log tape is typically complex.
- Often times System Utilities are used to read and interpret the log tape because of their complexity.
- Log tape captures all events that occur.

ANALOG DATA ISSUES

- First Issue = Sheer Volume of Data.
    - Massive amount of data generated.
    - A snapshot every millisecond.
    - 99.9% of little business value.
    - The same value is repeated over and over.
    - But the interesting data "hides" behind the tremendous volume.

- Second Issue = Important Metadata / Metaprocess Information is lost.
    - Metaprocess Information = Information that describes the data.
    - Metaprocess Information is more valuable than the actual Analog Data.
    - But analysts only keep the Analog Data and not the descriptor data

TRANSFORMING / CONDITIONING RAW ANALOG DATA

ONE WAY OF ANALOG DATA CONDITIONING = DATA REDUCTION

- In the past, the process of conversion was called Data Reduction and/or Data Compression.
- Data Reduction was to significantly reduce the amount of storage and the number of records that was required.
- This will reduce the amount of system processing.
- Some **Data Reduction techniques** are:
    - **Deduplication** = removal of masses of redundant data. Remove those data that has low probability of usage and place them somewhere less conspicuous.
        - Clustering = grouping similar and exact values of data.

        - Clustering is a form of data deduplication.

- **Excision** = removal of unneeded data.
    - Thresholding = Values above (or below) the threshold are stored. Everything within the boundaries of the threshold are ignored.
        - Thresholding is a form of excision.
    - Rounding = removing and rounding insignificant digits.
        - Rounding is a form of excision
- **Compression** = data packed very tightly.
    - The problem with compression arises when compressed data must be altered.
    - It is difficult to alter highly compressed data without incurring a high overhead.
- **Smoothing** = removing outliers.
- **Interpolation** = inferring values near to the value being created. That is, the "likely" value, had a value been found.
- **Sampling** = selecting a small subset of data that is representative of a larger set of data.
- **Encoding** = representing long strings with shorter strings.
    - Tokenization = a form of encoding.
        - Used effectively when there is a high degree of repetition in the data.

ANOTHER WAY OF ANALOG DATA CONDITIONING = FORMING DATA RELATIONSHIPS

- Data Relationships help to "connect" and give Analog Data meaning.

- Example of **Forming Data Relationships:**
    - 35.6 psi → Goodrich → July 20, 2016 → 16,500 miles
    - 36.1 psi → Bridgestone → Jan 5, 2013 → 85,980 miles
    - 34.6 psi → Goodyear → Oct 6, 2015 → 24,000 miles
    - 36.2 psi → Bridgestone → Nov 17, 2016 → 2,000 miles
- If we were just given the psi alone, it doesn't mean anything.
- If we connect the psi to the manufacturer name, there's more meaning.
- If we enhance the relationship by connecting to date and miles, even more meaning.

EXAMPLES OF APPLICATION DATA

- Application Data = Transaction Data
    - Sales data
    - Payment data
    - Banking checking data
    - Shipment data
    - Contract completion data
    - Inventory management data
    - Billing data
    - Bill payment data
- Application Data (compared to Analog / Textual Data) is probably the "cleanest" because it has been generated by an application.
- All the data in the application pond is uniformly structured and contains values that are relevant to the execution of some business activity.

DATABASE FORMAT OF APPLICATION DATA

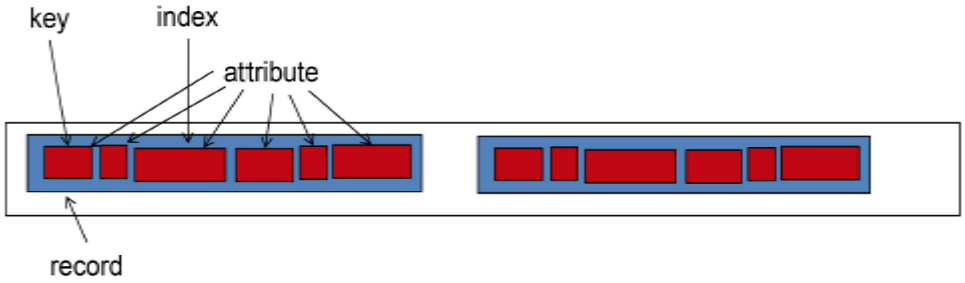


Figure 7: Application Data (Inmon 2016)

- Application records have a repeating uniform structure (Figure 7).
    - **Attributes** of the Application Data
    - One or more of those **Attributes** may be designated as a **Key**.
    - One or more of the **Attributes** can have an independent **Index**.
    - Details of Application Data are divided into **Records**.
    - **Records** have **Attributes** and some **Attributes** can be **Keys**, while other **Attributes** can be **Indexed**.
    - **Records** may have been shaped by a **Database Management System (DBMS)** application.
- Application Data enters the pond in a **Standard Relational Database (SRDB)** format.
    - A Relational Database is a digital database based on the **Row and Column** (Relational) format.
    - A software system used to maintain relational databases is a **Relational Database Management System (RDBMS)**.
    - Many RDBMS use **SQL (Structured Query Language)** for querying and maintaining the database. (Wikipedia 2020)
- Compared to Analog Data,
    - Analog Data = Just 1 very very long column

    - Application Data = Rows and Columns

- Just because data arrives in a **SRDB** format does not mean the advantages of that **DBMS** is carried into the pond.

- But once the data is in the pond, it is governed by whatever technology that is used to manage the pond, which most likely is **Not** a **SRDBMS**.

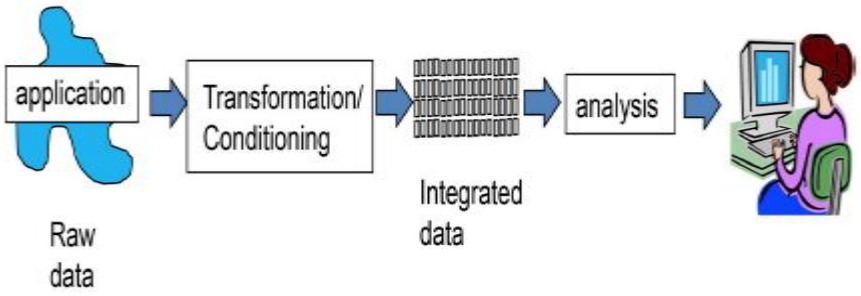


Figure 8: Integration Mapping (Inmon 2016)

- Integration Mapping is the method for Transforming / Conditioning Application Data.

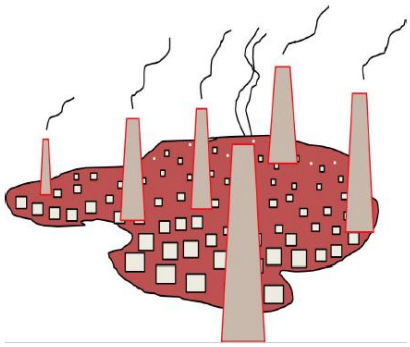


Figure 9: Data Silos (Inmon 2016)

- Application Data is notoriously unintegrated.
- Figure 9 shows Data Silos = unintegrated data in the data lake.
- Reading and interprating Data Silos is very difficult because each application is written in a different coding language.
- Every silo cannot communicate with other silos (even if properly tagged with metadata).
- Integration Mapping = a detailed specification of how data from one application can be meaningfully combined to another.
- Integrated Business Orientation = organizing data along the lines of the company.
    - Example: Customer / Product / Shipment / Order / Delivery.

- Application Data must be integrated and aligned with the business so that the analyst can make sense of the data.

AN EXAMPLE OF INTEGRATION

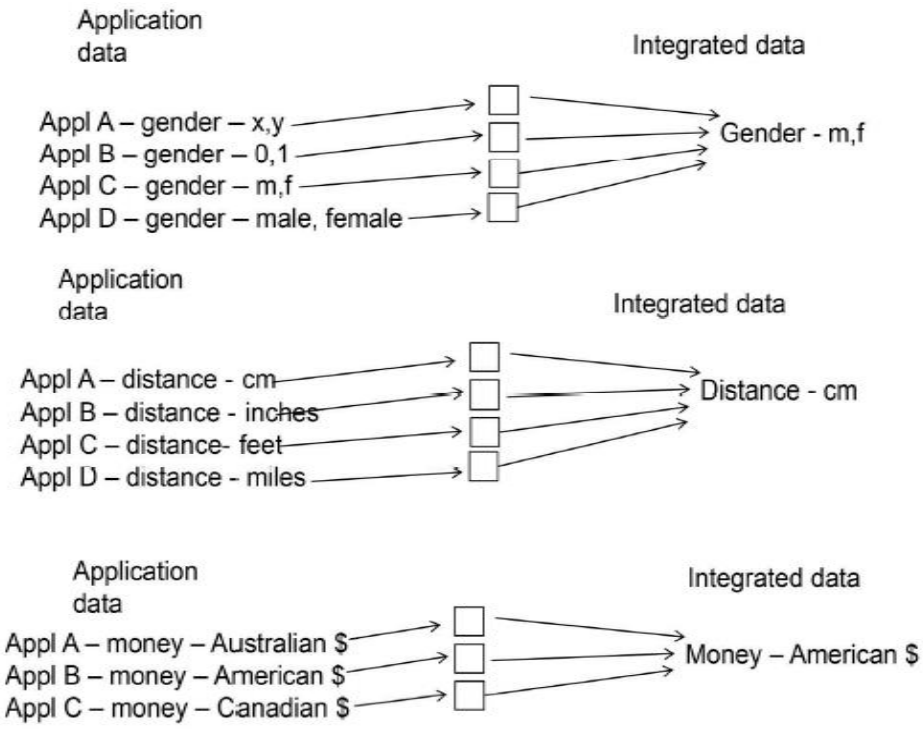


Figure 10: Example of Integration (Inmon 2016)

- Figure 10 shows an example of Integration.
- Integration = make the analysis consistent
    - Transformed into consistent definition of Gender – m,f.
    - Transformed into consistent measurement of Distance – cm.
    - Transformed into consistent currency of money – American $.

ANOTHER EXAMPLE OF SIMPLE INTEGRATION

- This example demonstrates a **Simple Pointer Relationship** – where the first Application points to the next.
- A **Customer Application** contains these names:
    - Bill Inmon / John Williams / Carol Renne / Georgia Burleson / Jeanne Friedman
- The **Ticket Database** shows:
    - Sat night 7:15 seat A12
    - Sat night 7:15 seat A13
    - Sat night 7:15 seat A14
    - Sat night 7:15 seat A16
- Upon integrating the two data above, the result is…
    - Bill Inmon – Seat A12 | Seat A13 – John Williams
    - Carol Renne – Seat A15 | Seat A15 – Georgia Burleson
    - Jeanne Friedman

ANOTHER EXAMPLE OF COMPLEX INTEGRATION

- This example demonstrates a **Complex Intersecting Relationship** – where two Application Data have an intersection and produces a third, independent Application
- **Existing Application Database** of Oil Company and Delivery Company:

App Data 1 - Oil Company	App Data 2 - Distribution Company
Standard Oil	Flying Horse Shipping
Conoco Oil	Akers Distributing

- An **Intersecting Database (App Data 1 + App Data 2)** could be a delivery being done:

- Delivery Code: AS15-YR
- From: **Standard Oil**
- To: 6534 Wolfensberger Road, Castle Rock, CO
- By: **Flying Horse Shipping**
- Amount: 2000 gallons
- Date: Sept 2

DATA MODEL

- **Data Model** = a ***High Level*** guidance as to how data should be related – related through entities and subject areas.
- But a ***Lower Level*** (more detailed) perspective accompanies the data model = Metadata.
- Data Model is very sophisticated because the analyst needs to know what changes have been made to the metadata over time – but the data model itself changes over time
  - Yet the Application Data Pond holds the data over a lengthy period of time.

TEXTUAL DATA POND

PLACES CONTAINING VALUABLE TEXTUAL INFORMATION

- Corporate contracts
- Corporate call center conversations
- Customer feedback
- Medical records
- Insurance claims
- Human resource records
- Insurance policies
- Loans applications
- Corporate memos

DIGESTING TEXTUAL DATA USING COMPUTERS

- Textual Data is not shaped into any Uniform Data; whilst Application Data is shaped into uniform records.
- But Computer systems are good at reading Uniform Data.
- They read one record, process it, then read another record that was in the ***same format*** as the previous record.
- Computer systems thrives on process repetition.
- That is why Textual Data is difficult for use for decision making

PROBLEMS WITH TEXTUAL DATA

UNSTRUCTURED DATA

- Textual data is called "unstructured data" because the text can take any form.
- For example, when a person is speaking, they can say anything in any fashion that they like.
- Usually the sounds make sense, but many variables can strip away the structure.
    - They could speak in riddles and parables.
    - They might use a different language.
    - Their speech may contain slang, vulgarities, be in a formal style or might even be an inside joke.
- Naturally, such text is extremely content dependent and not easily searched or processed by automated means.

CONTEXT

- Text without context is meaningless data.
- Suppose the text "court" appears.
    - Does court refer to a tennis court?
    - To a legal proceeding?
    - To the activities of a young man as he tries to lure a young lady as his mate?
    - Does court refer to the people surrounding royalty?
- If you are going to put text in the data lake, then you must also insert context as well, or at least a way to find that context.

INLINE CONTEXTUALIZATION

- Inline contextualization = how to identify the "context" of the text by examining the words surrounding it.

- Example:

```
Bill Inmon Signature
------------------------
signed by the leasholder
```

- The context here is → Bill Inmon is the leaseholder
- (identified because of the sentence below the line).

PROXIMITY

- Proximity = closeness of words
- Example: Denver Broncos won the Super Bowl
- The words "Denver Broncos" are taken to mean a professional football team.

ALTERNATE SPELLING

- Alternate Spelling = In England, the word "colour" is spelled color.

HOMOGRAPHIC RESOLUTION

- Homographic Resolution = Acronyms.
- But acronyms depend on context.
- Example:
    - A cardiologist → H.A = Heart Attack.
    - An endocrinologist → H. A. = Hepatitis A,
    - A general practitioner → H.A. = Head Ache.

ACRONYM RESOLUTION

- In the military, AWOL means absent without leave.

CUSTOM VARIABLE RECOGNITION

- In the US, the digits 999 999 9999 are interpreted to mean a telephone number.

TAXONOMY RESOLUTION

- When a document refers to a Volkswagen or a Honda, it is referring to a car.

DATE STANDARDIZATION.

- July 5, 1999 is the same thing as 1999/07/05.

TAXONOMIES AND ONTOLOGIES

Taxonomies are classifications of terms. Example:

- Taxonomy of Car → Honda / Porsche / Volkswagen / Ford / Toyota
- Taxonomy of Tree → Elm / Pine / Fir / Oak / Walnut

Ontology = a group of related taxonomies. Example:

- Taxonomy of Countries → USA / Canada / Mexico / Australia / South Africa
- Taxonomy of USA → Texas / New Mexico / Arizona / Colorado
- US is made up of states.
- Both "Taxonomy of Countries" + "Taxonomy of USA" = Ontology

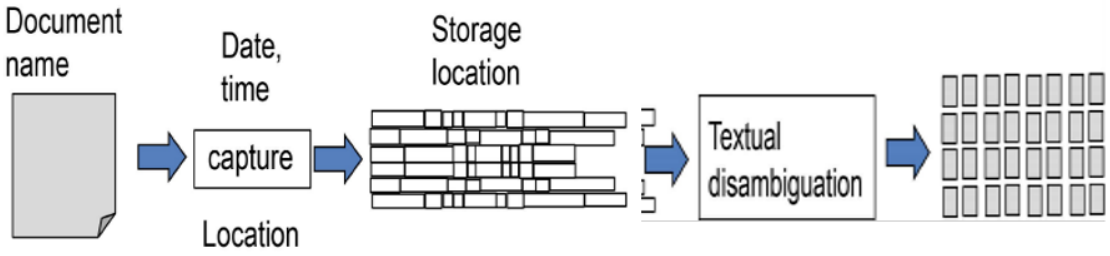


Figure 11: Textual Disambiguation Process (Inmon 2016)

- Figure 11 shows the Textual Disambiguation Process.
- Textual Disambiguation has two key steps:
  - Step 1: Text is **restructured into a uniform** format
  - Step 2: **Context identified** and attached to the text itself.

EXAMPLE OF TEXTUAL DISAMBIGUATION

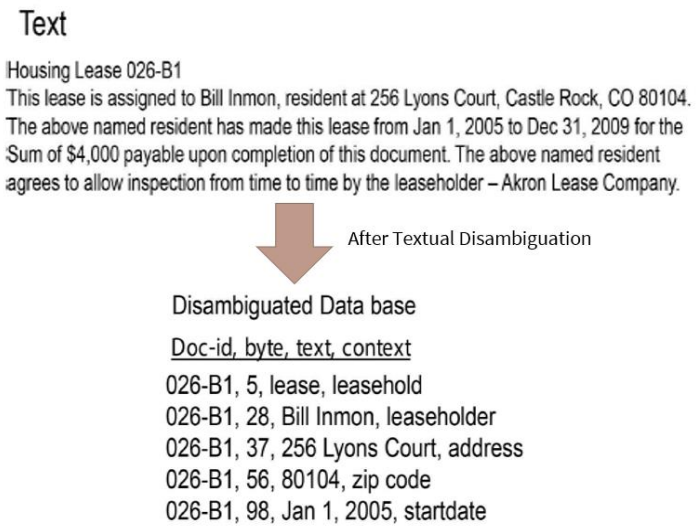


Figure 12: An Example of Textual Disambiguation (Inmon 2016)

SENTIMENT TEXTUAL DISAMBIGUATION

- Step 1: Create a Taxonmy of Sentiments (example Figure 13).

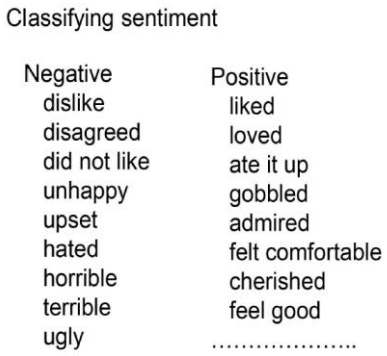


Figure 13: Taxonomy of Sentiments (Inmon 2016)

- Step 2: Match the Raw Text contents against the Sentiment Taxonomy.
    - Raw text may come from tweets, emails, documents.
- Step 3: Once a word matches a word in the taxonomy, inference is made on the sentiment.
- Step 4: The tone of the document can be gauged / weighted.
- Step 5: Document is placed back into database.
- Step 6: Analytical and visualization technology is applied to the database where multiple messages are analyzed.

EXAMPLE OF ANALYZING A DATABASE

- Example of 100,000 messages on Restaurant Feedback:
    - Item Taste: Too salty / too hot / too small portions.
    - Waiter: slow / bad attitude / very nice.
    - Cleanliness: Floor was wet / Table not wiped / Lights too dim.
    - Other topics etc etc.
- There are too many messages

- Restaurant chain runs its customer feedback through Textual Disambiguation.

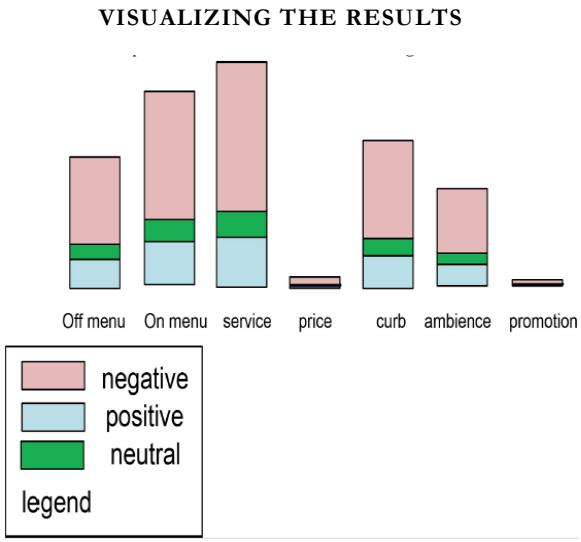


Figure 14: Restaurant Feedback Results (Inmon 2016)

NEGATIVE COMMENTS

- At first glance it appears there are a lot of negative comments.
    - But people are more inclined to message when there is a negative experience.
    - When a patron has a pleasant experience, he rarely feedbacks.
- A ratio of 85% is to 15% of negative to positive experiences is the normal expectation.
- If > 85% negative comments then something is wrong.
- If < 85% negative, that branch is doing something right.

COMMENTS ON PRICING + PROMOTIONS

- Figure 14 shows almost no comments about price → Management may not be charging enough for its food.
- Figure 14 shows almost no comments about promotions → Promotions are ineffective.

ARCHIVAL DATA POND

PURPOSE OF ARCHIVAL POND

- To have a place to store data that might have some future use
- To allow useless data to be removed from data ponds so that analysis in those data ponds can proceed in an efficient manner.

CRITERIA FOR STORING / REMOVING DATA FROM ANALOG / APPLICATION / TEXTUAL DATA PONDS

- Old data.
- Low usage.
- Must store due to litigation (legal needs).
- Must store because it is critical.

STRUCTURAL ALTERATION

- The purpose of Structural Alteration is for analysts to find data efficiently in the Archival Pond.
- Altering data occurs when data moves from Analog / Application / Textual Data Pond into Archival data pond (Figure 15).
- Data in Archival Data Pond has both metadata and metaprocess information attached directly to the raw data (Figure 15).
- For Analog Data Pond → Alter in a way of Data Reduction and Data Compression.
- For Application Data pond, → Alter in a way of classical ETL
- For Textual Data Pond → Alter in a way of Textual Disambiguation

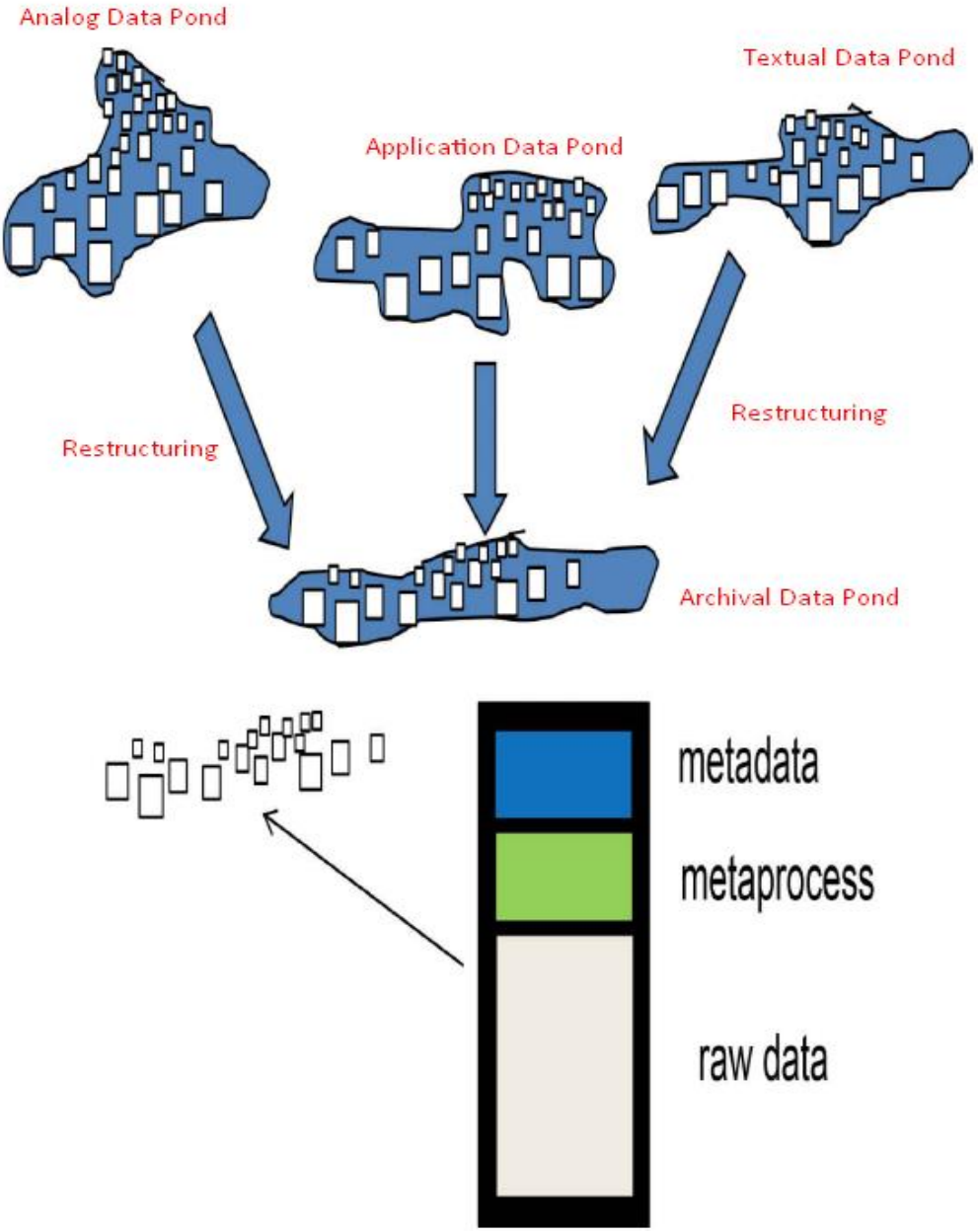


Figure 15: Structural Alteration (Inmon 2016)

- Step 1: Analysis = Past historical search. (Hill 2011)
- Step 2: Analytics = Future predictive modeling (Hill 2011)

ANALYSIS

- Analysis = Searching for data.
- 2 kinds of searches:
    - Search thru specific data → Example: Last medical record for Bill Inmon.
    - Search thru a group of data → Example: All records for people > 70 years old.
- Easy to search if…
    - Data is indexed
    - Data has been conditioned - easy to access and analyze.
    - Even after data has been found, it needs to be converted before it can be used
    - The qualifications for data are unclear.
- Tough to search if …
    - Data is hidden or disguised, such as
        - encrypted data
        - lurking behind a lot of mundane data points
    - Data is marked by very faint markers e.g. fictitious data.
    - The criteria for finding data is very unclear → Once you have found something, you are not sure it is actually the data you want
        - Machine Learning and Concept Search are dedicated to searching for data where the criteria for searches are murky.

ANALYTICS

THE MERE SORTING OF DATA

- Sorting data allows important data to surface and become obvious.

SUMMARIZING DATA

- Summaries of data bring to light data that is overlooked.

COMPARING DATA

- Contrasting to other sets of data often yields insight.

EXCEPTION ANALYSIS

- Finding outliers often lead to insight.

VISUALIZATION

- Visualization is popular because with a properly created visual setup, massive amounts of data can be depicted in such a fashion that important conclusions are immediately obvious.

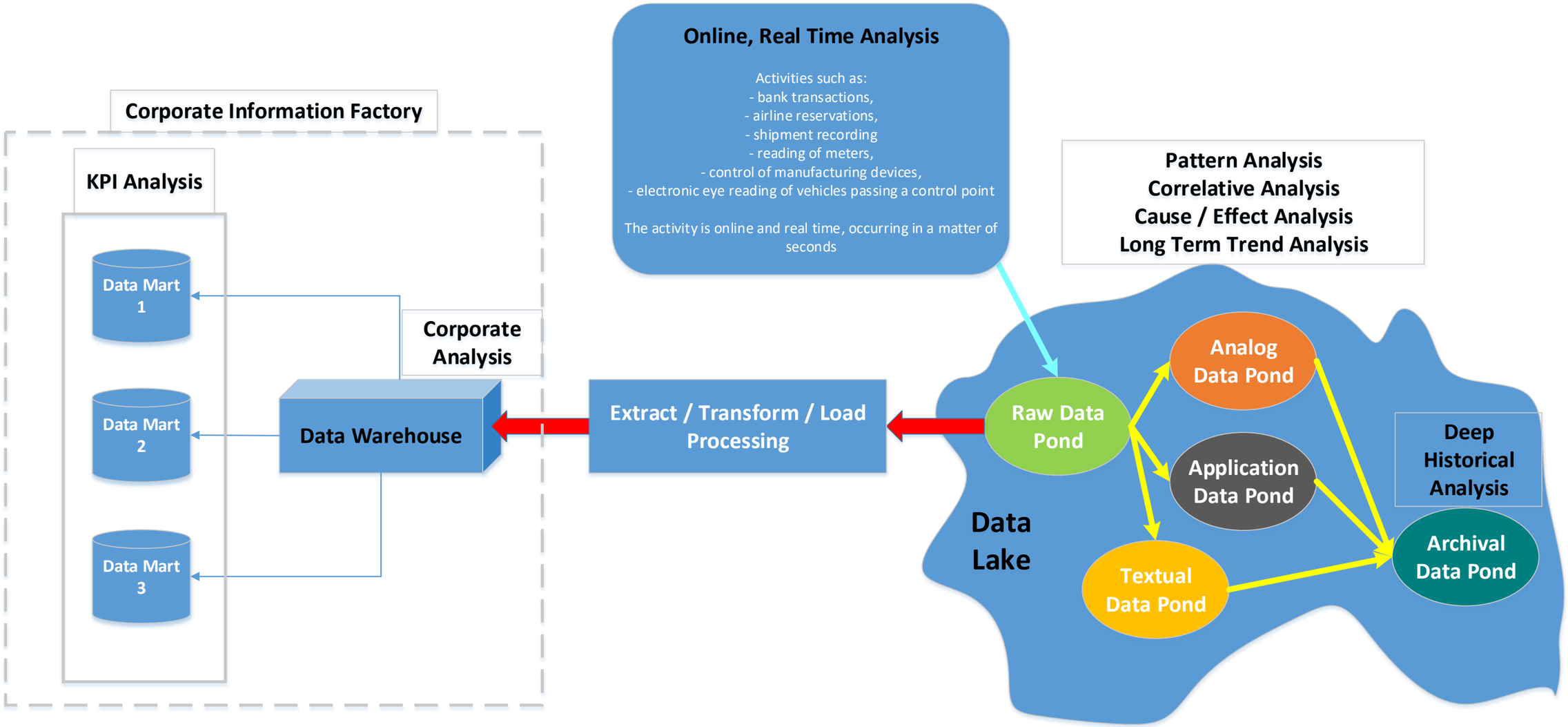


Figure 16: The Analytics Flow Within a Corporation

- Figure 16 shows the analytics flow within a corporation.

- It starts off with
    - "Online Real Time Analysis", after which it gets stored in the
    - "Raw Data Pond" … after which it can either go further down into the "Data Lake" or "Data Warehouse".
    - "Data Lake" is where Unstructured Data is stored. (because there is no Exchange / Transform / Load (ETL) processing.
    - "Data Warehouse" is where Structured Data is stored (because there is ETL processing)

INSIDE THE DATA WAREHOUSE

- Data Warehouse = Central corporate analytical location

- 3 to 5 years' worth of history is stored here.

- Analytical processing ranges from 5 minutes to 24 hours.

- In order to get data into Data Warehouse, it passes through ETL processing (data is transformed from Application State to Corporate State).

INSIDE THE DATA MARTS

- Surrounding the Data Warehouse are Data Marts.

- KPI analysis occurs inside Data Marts, typically on Departmental Basis.

- Marketing, sales, finance and so forth all have their own KPI analysis.

OUTISDE THE CORPORATE INFORMATION FACTORY (CIF) = ONLINE REAL TIME ANALYSIS (WITHIN THE APPLICATIONS)

- Various other analysis occurs outside the Corporate Information Factory (CIF).

- Analysis is detailed and immediate e.g.

- reading of meters,
- the control of manufacturing devices,
- the electronic eye reading of vehicles passing a control point.
- bank transactions,
- airline reservations,
- manufacturing control activities,
- shipment recording

CONFUSION BY VENDORS

ANALYSIS VS ANALYTICS

- Vendors confuse search with analysis (or rather Analysis vs Analytics).
- Vendors only sell Analytics Solution – not Analysis.
- They only sell one part – neglecting the other.
- Vendors always try to sell their solution as if it were the only solution.
- Vendors hate architecture because it lengthens their sales cycle
- Vendors make assumptions about data that simply are unrealistic
- In reality, vendors don't like anything except a sale.

BUSINESS VALUE IN ANALOG DATA POND

VALUE 1 = PREVENTING BAD CONSEQUENCES

Using Small Handful of Records

- Example: Manufacturer of Airbag for cars
    - Suppose an accident occurs where an airbag does not go off.
    - Investigator determines that the airbag was manufactured in March 1995 at the Phoenix, Arizona facility.
    - Investigator alerts all the car owners (who bought their car during this period) to have their airbags checked, thus avoiding a potential consequence.
- In this case, Analog Data was examined to find a handful of Analog Data/records that had potentially serious consequences.

VALUE 2 = IMPROVING CURRENT PROCESSES

Using Large Vistas of Data

- Looking across large vistas of Analog Data in a hurry.
- Example: Car airbag manufacturer management wishes to rethink how airbag is manufactured because of new technology.
    - Manufacturer looks at a massive amount of Analog Data to determine how many airbags have the older firing mechanism.
- Instead of looking for a few points out of many, analyst is looking for ***patterns of data*** which are manifest across many, many records.

BUSINESS VALUE IN APPLICATION DATA POND

VALUE = ACCOUNTING PURPOSES

Example 1: Locating Expense Receipt

- Looking thru the Application Data Pond to find a document – needed to prove to an auditor an expense item.

Example 2: Historical Shipment Costs

- In order to get a historical perspective on costs, management goes back to 1999 to calculate shipment costs.
- Calculation must be done using many, many documents.

BUSINESS VALUE IN TEXTUAL DATA POND

VALUE 1 = DOCUMENTATION OF PAST RECORDS

- Example: Important Price agreement is written on a paper letter.
- The organization searches the entire Textual Data Pond in order to find one document.

VALUE 2 = DETERMINING CUSTOMER SENTIMENT

- Knowing customer sentiment is an extremely valuable thing for the business.
- Customer sentiment is stored in the Textual Data Pond, expressed in many ways – through tweets, through emails, through other forms of narration.

BUSINESS VALUE HELD WITHIN PERCENTAGE OF RECORDS

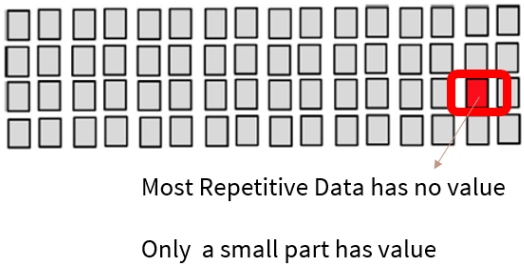


Figure 17: Small Data Point with Important Data

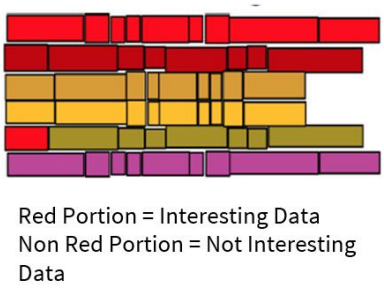


Figure 18: Percentage of Interesting Data Within Textual Data Pond

- Figure 17 shows that sometimes, only a small part of the Data has value.
- Figure 18 shows that sometimes, interesting data falls all over.
- Example: Telephone Calls.
    - Each phone call represents a customer's concerns or message.
    - The content of ***each*** phone call has real business value.
    - But when you look at the percentage of telephone calls made each day versus the total number of calls; the percentage is very low.
    - Perhaps the percentage is as low as .0000001%.
- Similarly, for log tapes, click stream records, and lots of other data, they too have very low percentages of business value.

DOCUMENTATION REQUIRED FOR BUILDING DATA LAKE

HIGH LEVEL DOCUMENTATION

High System Level Documentation (HSLD)

- HSLD shows the business analyst the general flow of data within the data lake/data pond environment.
    - How data **enters** the data lake and/or data pond
    - How data **flows** from one data pond to the next
    - How data flows into the Archival Data Pond.

LOW LEVEL DOCUMENTATION

Detailed Data Pond Level Documentation (DPLD)

- DPLD covers:
    - Metadata
    - Metaprocess information about the activities taking place in the data pond

Transformation Documentation (TD)

- TD =
    - The **criteria for entry** into the data pond
    - The **criteria for exit** out of the data pond.

HOW OLD IS THE DATA?

VERY FRESH DATA (FEW SECONDS OLD)

- Found in the Operational Environment

ONE TO FIVE YEARS OLD

- Found in Data Warehouse / Data Marts

ANY AGE (VERY YOUNG OR VERY OLD)

- Found in Data Lake
- The data lake is the original long-term carrier of data.

WHY STORED IN DATA LAKE RATHER THAN ELSEWHERE?

STATUTORY REQUIREMENTS

- Some data must be kept forever because of legal mandate → Thus stored in Data Lake.

CHEAP TO STORE THAN TO RECREATE THE DATA

- If it can be created electronically, it can be stored cheaply → inside Data Lake.

CURRENTLY, NO FORESSEABLE USE OF THE DATA…

- If its important to be created in the first place, it can be stored for future use → inside Data Lake.

DATA SECURITY INSIDE DATA LAKE

- Data Lake doesn't need as stringent a security as Data Warehouse.
- Because data in Data Lake is likely to be much older than data found elsewhere.

WHAT IF THE DATA IS NOT ANALOG / APPLICATION / TEXTUAL DATA?

WHERE TO STORE?

- If Data is **NOT** Analog / Application / Textual…. Do **NOT** place inside any of these pond

- Create a new pond – called the "Miscellanous Data Pond".
- Place this pond **Inside** the Raw Data Pond

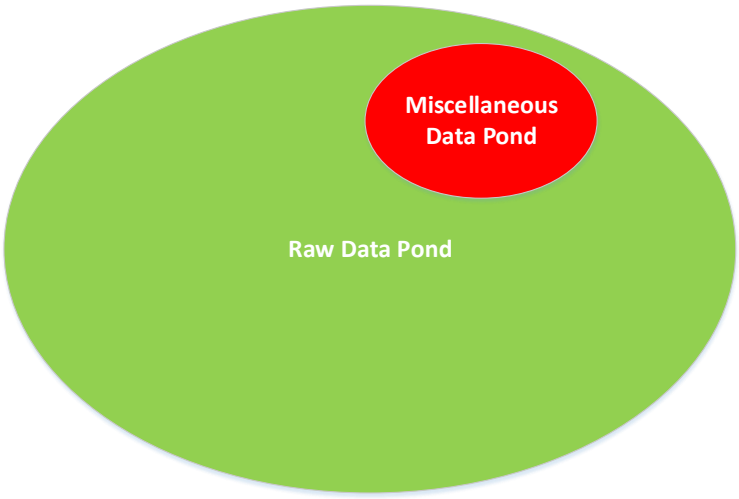


Figure 19: Carving Out a Miscellaneous Data Pond within a Raw Data Pond

- Data in the "Miscellaneous Data Pond" must be conditioned in order to support business analytical processing.

MUST THE FINAL DATABASE BE IN A RELATIONAL DATABASE FORMAT?

- No.
- The reason that most Databases are in Relational Format is because Analytical Packages (Statistical + Visualization) support Relational Database.

MUST ALL DATA PONDS BE USING THE SAME TECHNOLOGY?

- No.
- However, its cheaper to use the same technology everywhere.

HOW MUCH DATA SHOULD EACH DATA POND STORE?

- Depends upon the business goals and the business.

- An engineering firm or a manufacturing organization is probably going to have lots and lots of analog data.
- A telephone company is going to have lots of application data.
- And a marketing research firm is going to have lots of textual data.

CAN WE MOVE DATA FROM POND TO POND?

- No – even though it is technologically feasible.
- Because each pond has a fixed infrastructure e.g. Metadata definitions / Metaprocess definitions.
- Moving the data requires moving the infrastructure

CAN ANALYTICS BE CONDUCTED FROM MULTIPLE PONDS CONCURRENTLY / SIMULTANEOUSLY?

- Yes – but better not.
- Because analytics is usually restricted to a single pond – due to the type of data (Analog / Application / Textual).
- In order for analysis to be done across Data Ponds, there must be synchronization of Metadata – in order for the analysis to make sense.

Hill, C. (2011). "Analysis vs. Analytics: What's the Difference?". from https://www.1to1media.com/data-analytics/analysis-vs-analytics-whats-difference.

Inmon, B. (2016). Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump, Technics Publications, LLC.

Wikipedia (2020). "Bill Inmon." from https://en.wikipedia.org/wiki/Bill_Inmon.

Wikipedia (2020). "Data Lake." from https://en.wikipedia.org/wiki/Data_lake.

Wikipedia (2020). "Relational Database." from https://en.wikipedia.org/wiki/Relational_database.

ABOUT THE AUTHORS

William H. (Bill) Inmon (born 1945) is an American computer scientist, recognized by many as the father of the data warehouse. Inmon wrote the first book, held the first conference (with Arnie Barnett), wrote the first column in a magazine and was the first to offer classes in data warehousing. Inmon created the accepted definition of what a data warehouse is - a subject oriented, nonvolatile, integrated, time variant collection of data in support of management's decisions. (Wikipedia 2020)

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.