INDUSTRIAL INTERNSHIP REPORT

SUBMITTED BY: **SHIKHAR SINGH**

COLLEGE: **VELLORE INSTITUTE OF TECHNOLOGY, VELLORE**

COLLEGE REGISTRATION NUMBER: **16BCE2316**

PROJECT TITLE:

USER JOURNEY ANALYSIS, VISUALIZATION AND PREDICTION

PROJECT COMPLETED AT: **MPHASIS NEXT LABS, MPHASIS LTD.**

LOCATION: **BANGALORE, INDIA**

UNDER THE GUIDANCE OF:

INTERNSHIP MENTOR: **MR. ROHIT PATEL, SENIOR MANAGER, MPHASIS NEXT LABS.**

REPORTING MANAGER: **MR. JAI GANESH, SENIOR VICE PRESIDENT AND HEAD, MPHASIS NEXT LABS, BANGALORE.**

DATE OF SUBMISSION: **20-JUNE-2019**

Mentor Signature **Manager Signature**

TABLE OF CONTENTS

INTERNSHIP DETAILS

DATE OF JOINING: 13-MAY-2019

DATE OF COMPLETION: 20-JUNE-2019

LOCATION WHERE PROJECT WAS COMPLETED:

MPHASIS LTD., PARIN, BAGMANE TECH PARK, C.V. RAMAN NAGAR, BANGALORE-560093.

ORGANIZATION: MPHASIS LTD., BANGALORE

WING / DEPARTMENT: MPHASIS NEXT LABS, PARIN, BAGMANE TECH PARK, C.V. RAMAN NAGAR.

ACKNOWLEDGEMENT

Firstly, I am grateful to Mphasis Limited for granting me an internship project. I would like to thank Mr. Jai Ganesh (Senior Vice President and Head, Mphasis NEXT Labs, Bangalore) for giving me this opportunity to take up this internship at Mphasis Next Labs, Bangalore. I would also like to thank my internship mentor, Mr. Rohit Patel (Senior Manager, Mphasis NEXT Labs, Bangalore) for his expert guidance and constant support. It was due to his patience and help that I was able to complete my objectives. I am also grateful to the entire NEXT Labs team for counselling me and enabling me to overcome all the challenges I faced. Lastly, I offer my utmost gratitude to my college, VIT-Vellore, for encouraging me and giving me a platform to pursue this internship opportunity.

SYNOPSIS

The project I have undertaken deals with business processes and how we can analyze, visualize and predict the activities a user takes in his/her journey from start to end of a process. Such an analysis helps enterprises take business decisions effectively and plays an active part in monitoring a businesses' growth.

The objectives of this internship project are listed below:

- To generate an accurate and functional visualization of the process journey.
- To extract process level information and perform path analysis from raw event data stored in a graphical database.
- To use contextual information of the user to predict the next step in the user journey.

An accurate and functional visualization of the process map is crucial to decipher the root cause analysis and bottlenecks at a glance. A flexible data store, such as a Graph Database would greatly optimize the data retrieval and data processing aspects. Next Step predictions are necessary as they aid important business decisions. These are some of the benefits that this project aimed to provide.

I was able to achieve user journey visualizations using R scripting language, after researching and performing a comparative analysis on all the R network visualization ecosystems. I used Neo4j Graph Database to formulate a structure for the raw event data and Cypher Query Language for insights and path analysis. Next Step Prediction was done in Python.

Some of the challenges I faced were, lack of proper ecosystems for the exact required visualizations in R or JavaScript, Integration of different technologies into a single application and modelling event-log data for Analysis and Prediction. I was able to overcome these challenges and, in the process, learnt new technologies like R-network visualizations, d3.js, vis.js, r-shiny, neo4j and Cypher.

Besides the invaluable experience, I have gained a lot from this project. Following are the key aspects I have learnt during this internship project.

- I have an active command over graph databases and can model any sort of data and perform analytics.

- I have learnt how to accurately visualize network data, and how to integrate it with a web application.
- I have learnt about the various process mining techniques and how they mine event-data.
- I have learnt to model raw event-log data and perform predictive analysis.
- Apart from this, I have learnt the use-cases where such a user journey analysis becomes necessary and what sort of inference the companies are looking for from their process data.

The project was divided into three phases:

- User Journey Analysis
- User Journey Visualization
- User Journey Prediction

Each of these phases are described in detail in the forthcoming pages.

PHASE 1 : USER JOURNEY ANALYSIS

CONTEXT

The optimization of data storage is critical in data retrieval processes in every large enterprise. The structure of the database determines the level of complexity in acquiring and filtering the database. A database management system stores, organizes and manages a large amount of information within a single software application. Use of this system increases efficiency of business operations and reduces overall costs.

In this project, a thorough examination of the Graph Database Structure was conducted in context of business process management. I used a Graph Database to capture the inherent sequential structure of event-logs generated by business processes. Event log data is highly connected, and the sequence of operations plays an important role in data analysis.

OBJECTIVES

The objectives of the project are listed below:

1. Database Structure:
    - Model Raw Event-data and feed into Graph Database. {extract node(entity) fields from event-data}
    - Determine and create relationships between entities in the Graph database.
2. Database Querying and Analysis:
    - Identify Unique common user journeys and how many times they occur. {paths}
    - Determine relationships between these journeys and the corresponding users that embark on them.
    - Perform path analysis.
    - Generate a process aggregation and display the frequencies of each transition.

GRAPH DATABASES

What are Graph Databases?

A graph database is a database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data. A key concept of the system is the graph (or edge or relationship). The graph relates the data items in the store to a collection of nodes and edges, the edges representing the relationships between the nodes. The relationships allow data in the store to be linked together directly and, in many cases, retrieved with one operation. Graph databases hold the relationships between data as a priority. Querying relationships within a graph database is fast because they are perpetually stored within the database itself. Relationships can be intuitively visualized using graph databases, making them useful for heavily inter-connected data.

Graph DB vs Relational Database Systems

RDBMS

- Works well when data is well structured and not join intensive.
- When data is more related, carries out complex join queries which require a lot of processing time and is expensive to run.
- JOINS are computed for every query.
- To structure event data and produce insights would require a lot of code in SQL.
- Relationships are set in the form of foreign key, primary key pairs
- No Graphical Visualization.

GRAPH DB

- Works well with data which is highly connected. Best for relationship networks
- Join index lookup performance can be avoided altogether, finding link between entities become as simple as walking through nodes.

- Establish relationships once at time of data modelling instead of computing them every time.
- Cypher Querying is much smaller and faster.
- Relationships are stored in the form of entities themselves.
- Visualization of DB In the form of VisNetwork.

Process Map Generation

Currently process flow maps or graphs are generated through bupaR ecosystem in R. My aim was to generate a similar graph with greatly enhanced visualization capabilities with the Graph Database at the backend.

NEO4J GRAPH DATABASE

The graph database used is Neo4j. Neo4j is a graph database management system developed by Neo4j, Inc. Described by its developers as an ACID-compliant transactional database with native graph storage and processing, Neo4j is the most popular graph database according to DB-Engines ranking, and the 22nd most popular database overall.

Neo4j is available in a GPL3-licensed open-source "community edition", with online backup and high availability extensions licensed under a closed-source commercial license. Neo also licenses Neo4j with these extensions under closed-source commercial terms.

Neo4j is implemented in Java and accessible from software written in other languages using the Cypher Query Language through a transactional HTTP endpoint, or through the binary "bolt" protocol.

DATA SET USED

For this project, the dataset I used is the raw patients event log data. The fields of patients database are:

- event_id {Unique identifier of the event}
- handling {The activity being performed}
- patient {Unique identifier of each case}
- employee {Unique identifier of each resource}
- handling_id {Unique identifier of each activity}
- registration_type {type of function in the activity: start or complete}
- time {timestamp of activity, for the given registration_type}
- . order {order of occurrence of events}

DATA PRE-PROCESSING

Modelling of the database was done with respect to the entities involved. In this case the entities were Cases(Patients) and the activities. These entities were represented as nodes. For each node I extracted a dataset from patients.csv. For simplicity, I selected the first 25 cases from patients event log.

The fields involved in Cases entity dataset were:

- case_id (Patient_id)
- case_name (Patient Name, No case_name in patients.csv so duplicated case_id's as case names)

The fields involved in Activities entity dataset were:

- event_id {Unique identifier of the event}
- activity_name {The activity being performed}
- case_no {Unique identifier of each case}

- resource_id {Unique identifier of each resource}
- activity_id {Unique identifier of each activity}
- start_time {timestamp of activity, for type = 'start'}
- complete_time {timestamp of activity, for type ='complete'}
- next_activ_id {next activity given a case (extracted from the dataset using date and time properties)}

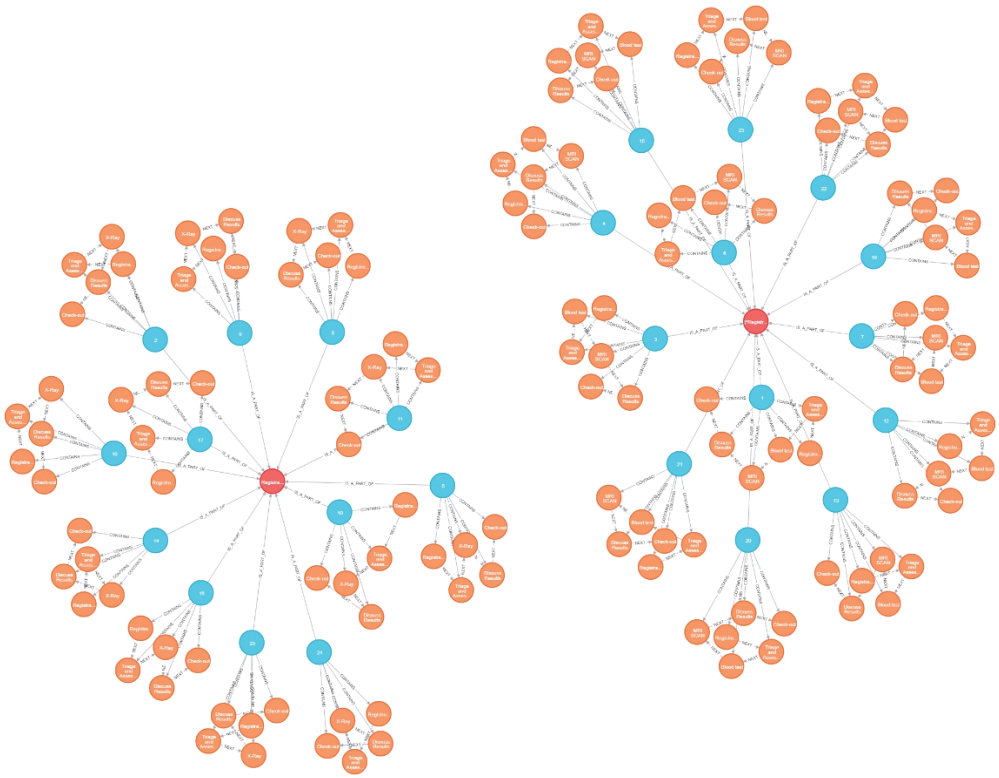
DATABASE STRUCTURE

NODES: ENTITIES

- CASE
- ACTIVITY
- TRACE

EDGES: RELATIONSHIPS

- CASE – {CONTAINS} -> ACTIVITY
- ACTIVITY – {NEXT} - > ACTIVITY
- CASE - {IS A PART OF} - > CASE



Structure of Graph DB created

INSIGHTS AND PATH ANALYSIS

The constructed Graph database can be used to generate insights and analyze user journeys.

Process Map Generated

Insights Captured:

- The cases in which an activity occurs.
- All the activities of a case.
- Given a trace, all the cases of which it is a part.
- Given an activity, what activity comes next and with what frequency?
- What journeys are taken from a given starting point and with what frequency?
- Given two activities, what is the number of activities between them and how often these paths occur?
- all the user journeys taken along with frequency.
- Aggregate user journey for all cases

Other applications and Use-Cases of Graph Database:

- Recommendation System
- Fraud Detection
- Network and IT operations

RESULTS

I was successfully able to capture the raw event data in the database structure and able to extract along with their frequencies the inherent paths of all the cases. Aggregating all these paths I was able to generate a process map and display the KPIs for each activity, case and trace.

PHASE 2: USER JOURNEY VISUALIZATION

CONTEXT

A lot of businesses deal with highly connected data. Social networking sites like Facebook, Twitter etc. have millions of users and all of them are connected to each other through different channels, ultimately forming a network. In such a scenario, it would be useful if we can visualize these networks in the form of graphs. Such a visualization would facilitate a better understanding of data elements and how they are exactly connected to each other. Moreover, the graphs generated can be used for various analytical purposes. Social Network Analysis which is the process of investigating social structures using concepts of graphs is one domain which benefits highly from an accurate and descriptive visualization. Centrality measure, chunking, degrees etc. are metrics that can be observed from these graphs and support business decisions.

In the case where the network describes a user journey (from start to end) in a process, an accurate and informative visualization becomes more crucial. A User Journey can be defined as the path a user takes from the starting to the end of a process. In terms of a business process, an aggregate of the paths followed by customers can be used in process flow discovery. The flow established, is used for data analyses such as Risk Point Analysis, Bottle-Neck Analysis etc. An accurate user-journey graph would assist the business in visualizing the results of these analytical practices through the graph itself.

OBJECTIVE

The objective of this phase of my project was to generate a visualization for event data (which is in the form of an event-log). Currently, this can be achieved using the bupaR ecosystem in R. Although, the user journey graph generated using bupaR has the following limitations:

- Static.
- Non-Interactive with User.
- Tough to decipher for large event-logs.
- No display of KPI(Key Performance Indicators) through the graph.

The aim was to visualize the user journey graph containing the following features:

- Dynamic directed graph.
- Node and Edge Selection by user.
- Events supported(on-click/on-hover).
- Displays node and edge information on-click/on-hover triggered by events.
- Supports replay token animations.
- Visualization based on JavaScript, with corresponding R-package which binds the JS libraries. This ensures easily deployment on a web-app.

RESEARCH

Process Map :

As stated in the objectives, the visualization had to be in JavaScript. Since the process maps were being generated in R, several R-bindings for JavaScript libraries were available. These R-bindings ensured that using the corresponding R package for the JS library, the graph could be visualized in R itself. The R-packages with their corresponding JS libraries are listed below:

- networkD3: d3.js
- igraph: igraph.js
- visNetwork: vis.js

Approach 1: D3.js

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

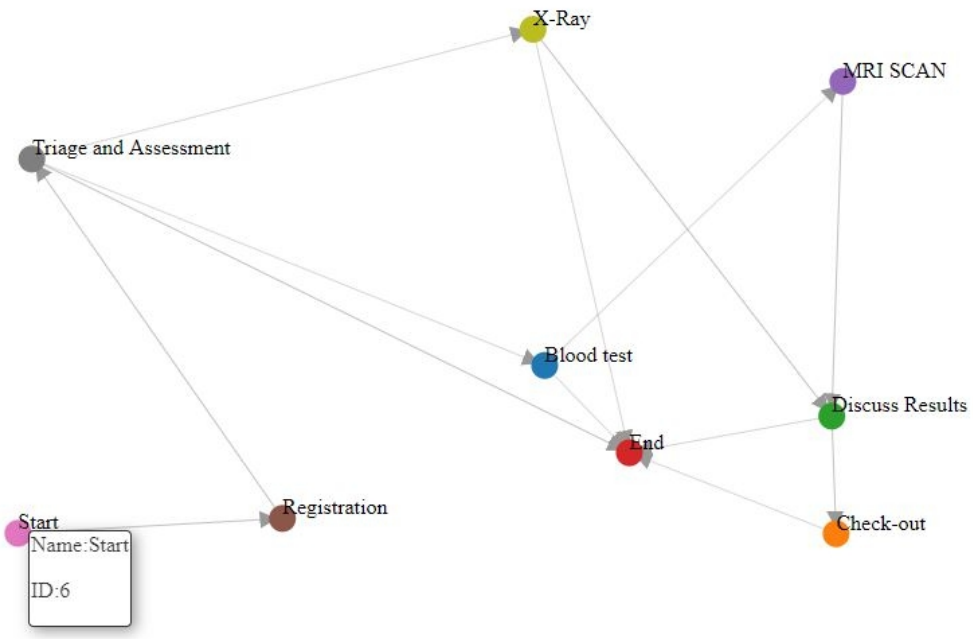
My first approach was to use the d3.js JavaScript Library to visualize the process map.

Objectives fulfilled through this approach:

- ☑ Dynamic directed graph.
- ☑ Node and Edge Selection by user.
- ☑ Events supported(on-click/on-hover).
- ☑ Displays node and edge information on-click/on-hover triggered by events.
- ☑ Visualization based on JavaScript, with corresponding R-package which binds the JS libraries. This ensures easily deployment on a web-app.

Challenges Faced were:

- Development time was excessive
- A lot of data formatting required in JSON file being fed to d3.js
- SVG elements needed to be created manually.
- Visualization was interactive, but functionality was limited.



D3.js process map

Approach 2: Igraph package

Routines for simple graphs and network analysis. It can handle large graphs very well and provides functions for generating random and regular graphs, graph visualization, centrality methods and much more.

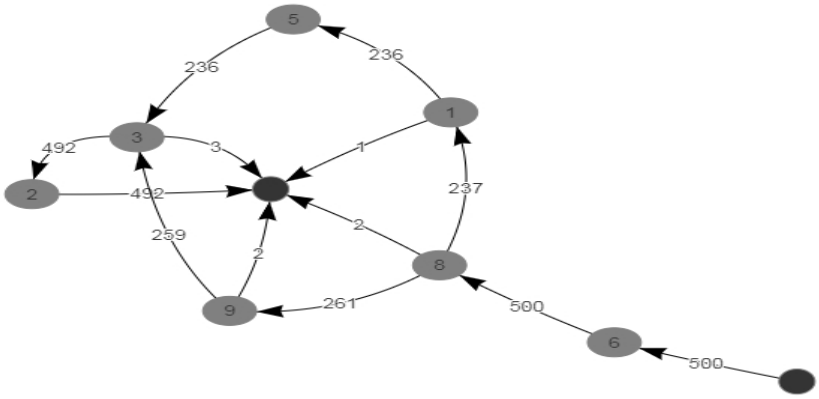
Objectives fulfilled with this approach:

- ☑ Dynamic directed graph.
- ☑ Node and Edge Selection by user.
- ☑ Visualization based on JavaScript, with corresponding R-package which binds the JS libraries. This ensures easily deployment on a web-app.

Challenges Faced were:

- No animation support
- No interactivity support

- Visualization not flexible to user.



Igraph Process Map

Approach 3: visNetwork

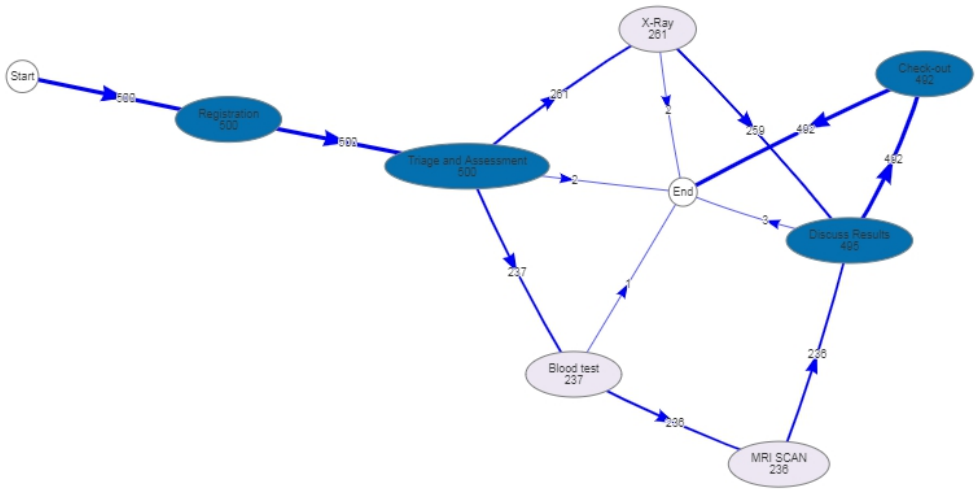
visNetwork is an R package for network visualization, using vis.js JavaScript library. It is based on HTML widgets and so it is compatible with R-Shiny. It proposes all the features available in vis.js API.

Objectives fulfilled through this approach:

- ☑ Dynamic directed graph .
- ☑ Node and Edge Selection by user.
- ☑ Events supported(on-click/on-hover).
- ☑ Displays node and edge information on-click/on-hover triggered by events.
- ☑ Supports replay token animations.
- ☑ Visualization based on JavaScript, with corresponding R-package which binds the JS libraries. This ensures easily deployment on a web-app.

Challenges Faced were:

- Animations were not in the expected format.



VisNetwork Process Map

Animations:

A useful addition to any User Journey graph would be the replay token animation. This animation is of the sort that each token represents a user. The token begins its journey at the start node and finishes at the end node. The transit time of the token from one node to another depends on the idle time between activities. Such an animation is possible using the ProcessanimateR ecosystem in R. Although its visual effect is limited by the underlying process map it runs on.

Challenges faced with Animating the graph:

- No ecosystem used supports the animations of this sort.
- Zoom-in/Zoom-out animations and node connection animations are available but not useful.

vis.js - vis.animate.traffic()

Vis.js does contain a function which generate tokens and traverse the edges of the graph on hovering over the node. Using this function, I was able to play an on-hover animation which caused the movement of tokens from one node to another in the visNetwork Process Map

Challenges occurred using vis.animate.traffic():

- The tokens generated merely traverse the edges.
- These tokens are not assigned any frequency.
- They do not run according to idle and processing timestamps.

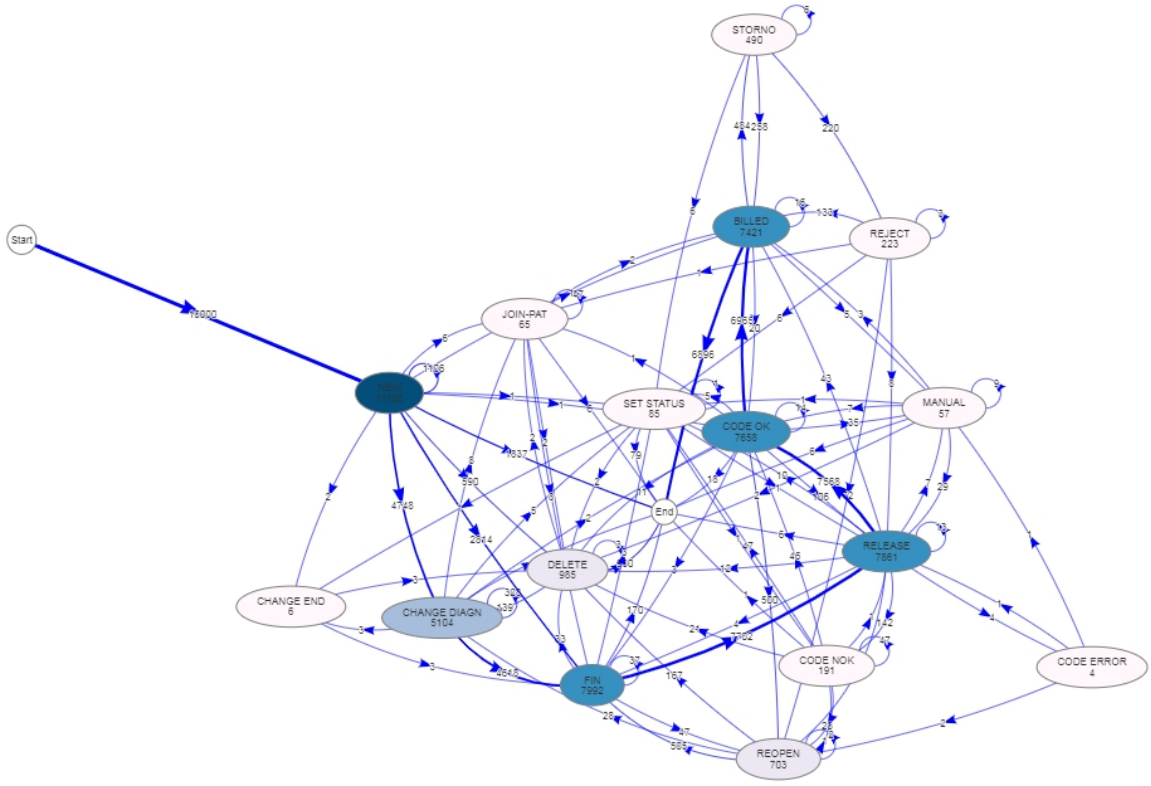
FUTURE ENHANCEMENTS:

Animating the graph in the required format to accurately display User journey flow. This can be achieved by enhancing the vis.animate.traffic() function to consume processing and idle timestamps along with the number of nodes required to flow between two particular nodes.

METHOD SELECTED

The method selected I selected was visNetwork as it fulfilled almost all the objectives of the graph visualization. It also gave way to future enhancements that could be done to achieve animations on a dynamic and interactive process graph.

RESULTS



VisNetwork Process Map

This process map was successfully deployed on an R-Shiny application and supported all interactions through the web interface.

PHASE 3: USER JOURNEY PREDICTION

CONTEXT

Predictions play an important role in business decisions such as resource allocation and process optimization. A user has a lot of attributes associated with him/her which forms a detailed description about him. The idea was to use these attributes of the user to predict what would be his/her most likely user journey.

OBJECTIVE

The objective of this phase of my project was :

- to predict next activity for a user given his/her inherent attributes (contextual variables)
- To test the accuracy of tree-based classification for this prediction and report the accuracy.
- To improve the predictions by selecting the relevant contextual variables to be applied.

DATA USED

The data used was provided the ninth International Business Process International Challenge which was co-located with IPMC this year (2019). This challenge provides participants with a real-life event log and challenges them to analyze these data using whatever techniques available, focusing on one or more of the process owner's questions or proving other unique insights into the process(es) captured in the event log. For the BPI Challenge 2019, the data was collected from a large Multinational Company in the Netherlands in the area of coatings and paints and it contained the purchase order data for some of its 60 subsidiaries

DATA PREPARATION

The raw data consisted of over 1.5 million records. My objective was to order the entire data by the cases and assign the next activity field to each activity of each case. This I was able to do using pandas package in python and using data. Table package in R.

MODEL USED

The model I selected was the Random Forest Classifier. I decided to use a tree-based approach as it handles missing values and maintains accuracy of large data. Moreover, the larger the number of trees, the lesser would be the overfitting in the model. Random Forest also has the capability of handling large datasets with high dimensionality.

CONSIDERATIONS

- Encoding the contextual variables using Label Encoder, OnehotEncoder and the Integer encoder. The encoding ensured that continuous variables were assigned higher feature importance.
- A single level of a categorical variable had to meet a very high bar in order to be selected for splitting early in the tree building. This degraded predictive performance.

RESULTS

I was able to predict the next activity using Random Forest Classifier. The accuracy of this classification came to be about 65% when trained on about 90 cases.

FUTURE ENHANCEMENTS

- Using h2o package in R with Random Forest which directly inputs categorical variables, to bypass the encoding process.
- Modelling the entire dataset to train the model adequately.
- Testing other classification models and performing comparative analysis.

ANNEXURE

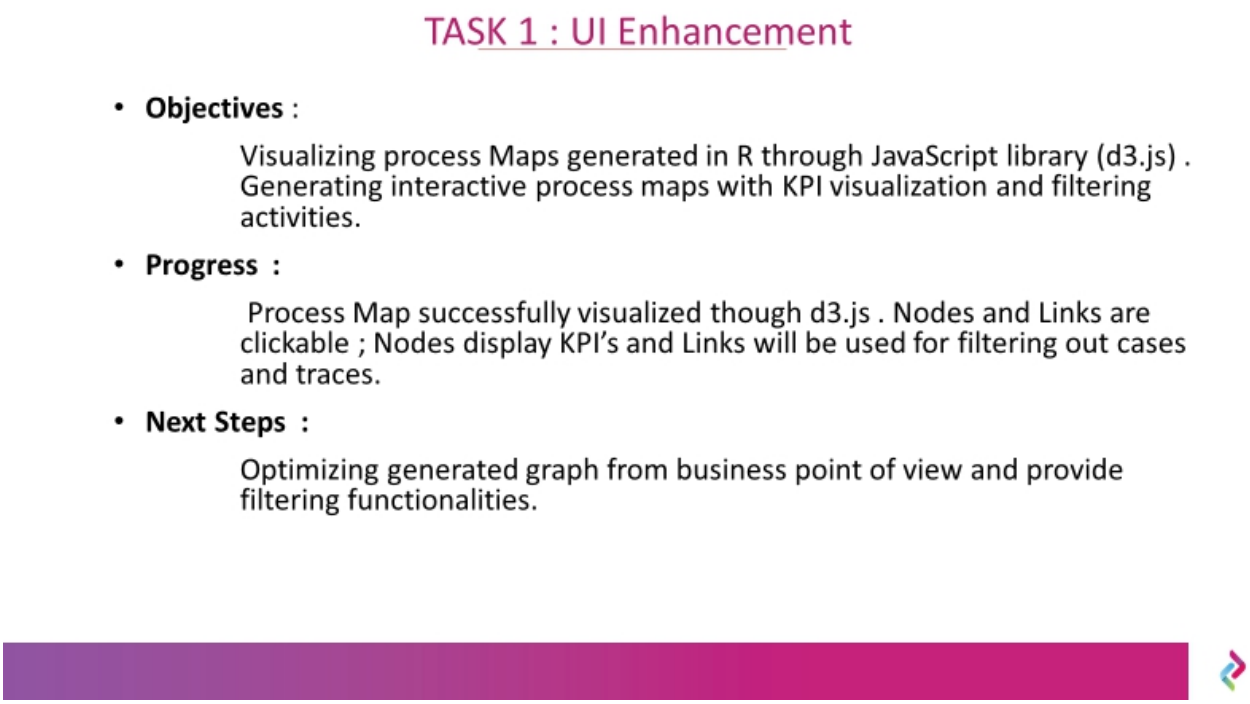
PRESENTATIONS GIVEN:

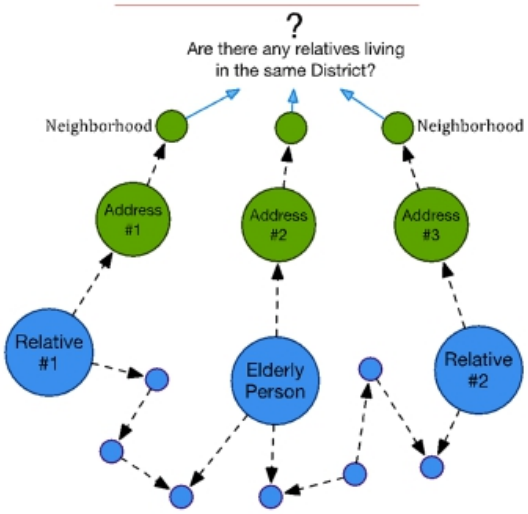
Presentation 1:

Date Of Presentation : 24 - MAY – 2019

Audience : Mphasis, NEXT LABS Team.

Slides :





Discovering Elderly People without any relatives living nearby.
Not possible at all with SQL

Task 3 : Prediction Analysis

- **Objectives :**

  Decision Tree/LSTM based next action, end point and throughput time for open user requests.

- **Progress**:

  WIP

- **Next Action**:

  Solution Design, Data Gathering and Algorithm Development.

Presentation 2:

Date Of Presentation: 14 - JUNE – 2019

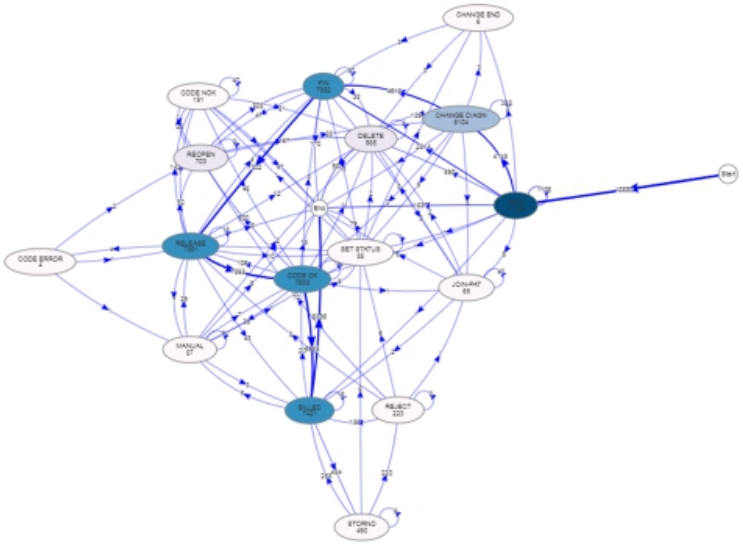
Audience : Mphasis, NEXT LABS Team.

Slides :

Approach and Research Work

- D3.js JavaScript library , R-package : networkD3. method discarded : not many high level abstractions in JavaScript that supported visualization, SVG elements has to be created manually, Visualization interactive but functionality was limited .
- I graph Package in R, explored visualization , method discarded as it lacked implicit features which helped in graph interactivity, fundamental to the project.
- Ndtv Package in R, supported animation in R such as node-connections with time , although visualization of interactive graph was limited and animation was too not of the desired format.
- Ggplot was also tested, yet it was suited to more of bar-graph , scatter-plot visualizations, than networks. Gganimate method was also explored, but that contained basic animations not required for this project.
- VisNetwork Package in R , which provides R-binding for vis.js JavaScript library. Visualization was of the desired format. Nodes were clickable, and returned KPI's on click. It is the latest package for network visualization . .Successfully integrable with rshiny and other graph platforms like gephi.

Enhanced visualization (hospital_billing)



- Clearer Representation
- Zoom/in Out Capabilities
- Nodes onclick display data related to activity
- Edges on click display source and target activities
- Force simulated
- Based on a JavaScript library, can be integrated easily to a web application.
- Can be deployed and edited through rshiny as well.
- Neighborhood nodes can be selected to show immediate flows.

Animations on Process Map

- **Objective**: Process map Replay - Tokens flowing between activities , the duration of flow between two activities must be equal to idle time between said activities and tokens must stay in the activity through the processing time.

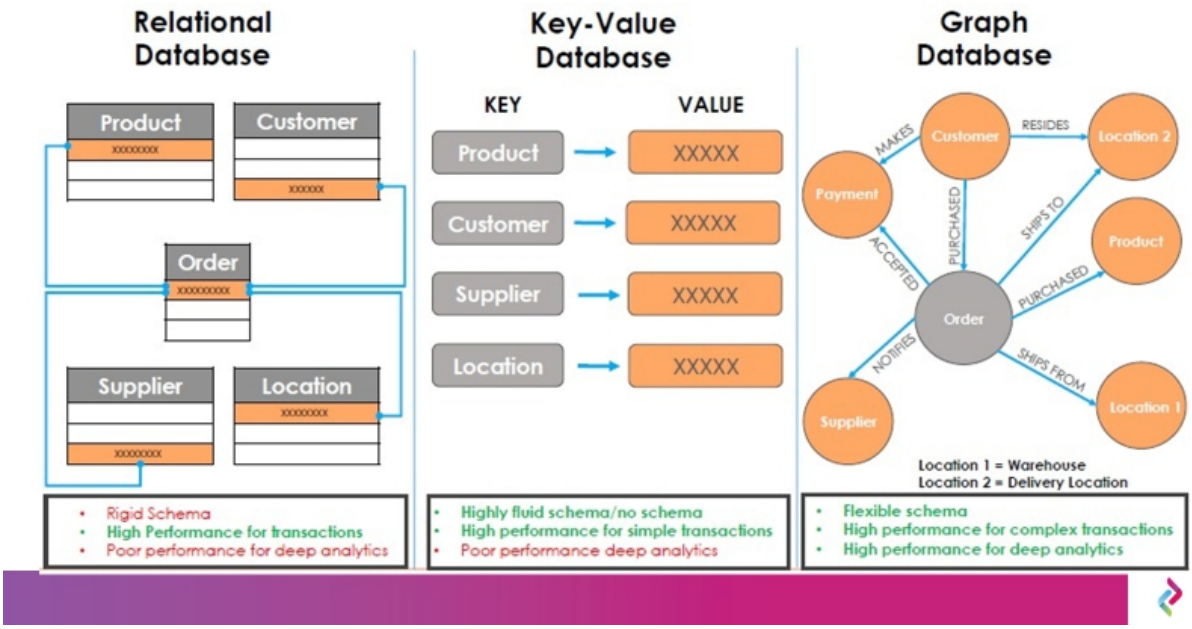
- **Challenge**: None of the network visualization packages had animation options which were similar to the objective.
- Ndtv3 and gganimate did show some animation capabilities, but were very limited.

- **Approach1:** Researched on ProcessAnimateR package, Renders SVG object which was not being generated by Visnetwork, that generates a canvas object instead.

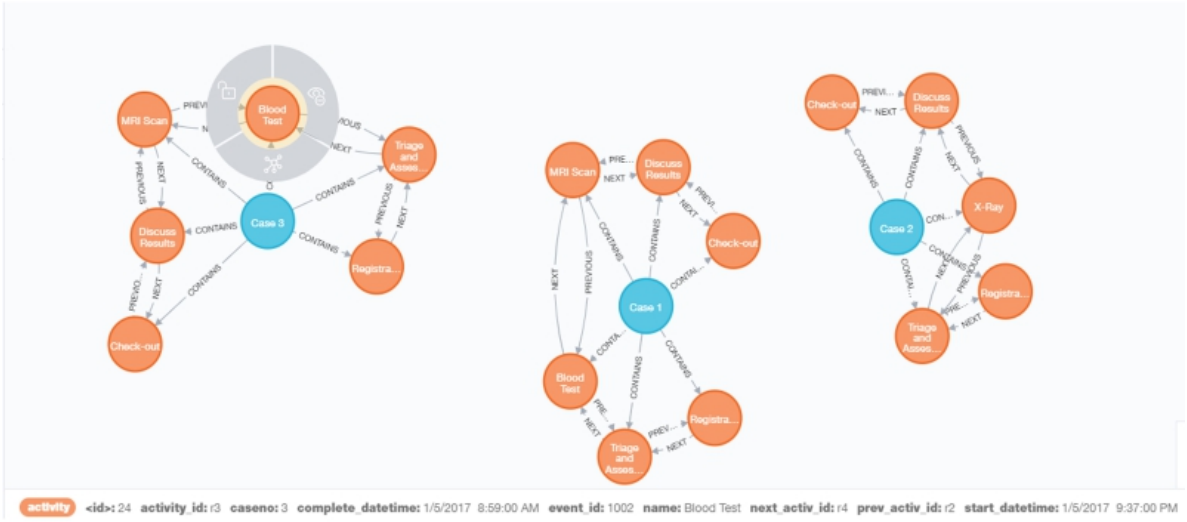
- **Approach2:** Researched on function vis.AnimateTraffic() – challenge – tokens travelling between nodes, do not follow the time constraints, merely run from source to destination. Need for inclusion of time and duration parameters.

- **Result:** Due to time constraints, animations on process map were put on hold and will be resumed later.

Database Comparison



STRUCTURE of Graph DB visualization



Questions answered and visualized

1. What all cases in which a particular activity occurs?
2. Which all activities performed by a case?
3. Given an activity, which is the next activity ? Case Level and Trace Level
4. Given an activity, which is the previous activity? Case Level and Trace Level
5. Given a trace, what activities it contains and in which order?
6. Given an activity ,which all traces does it belong to?

NEXT STEPS :

1. Query the data base to perform path analysis :

   What are the most common paths that end at a particular activity?
   What are some common user journeys?

2. Add Resource Nodes and define relationships , to set up resource->activity mapping.

REFERENCES

- Graph Database : https://neo4j.com/
- Neo4j with event data : https://snowplowanalytics.com/blog/2017/07/17/loading-and-analysing-snowplow-event-data-in-Neo4j/
- Visnetwork : https://visjs.org/
- D3.js : https://d3js.org/
- Process Mining course : https://www.coursera.org/learn/process-mining