RESEARCH WORK – SHIKHAR SINGH

Contents:

1. **Publication:**
   Human Computer Interaction: Game Developed in Python using Gesture Recognition
   (International Journal of Engineering Research & Technology)

2. **Research project documentation:**
   User Journeys: Analysis, Visualization and Prediction
   (Mphasis NEXT Labs)

PUBLICATION

Human Computer Interaction: Game Developed in Python using Gesture Recognition

(International Journal of Engineering Research & Technology)
November 2019

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

Human Computer Interaction: Game Developed in Python using Gesture Recognition

Shikhar Singh
SCOPE School, Vellore Institute of Technology,
Tamil Nadu , India.

Dr. Swarnalatha P
SCOPE School, Vellore Institute Of Technology
Tamil Nadu, India

***Abstract*— This paper has been taken up to demonstrate the an emerging field of Human Computer Interaction i.e. Gesture Recognition. Gesture recognition has found its way into many applications ranging all across from basic home automation to navigation and gaming. In this paper, I have explored the capabilities of gesture recognition by demonstrating the classic Snake game as a gesture controlled snake . We use Computer vision techniques and the OpenCV library to achieve our results, which are similar if not more fluid then the original mechanical version that makes use of the arrow keys. The key aspects of the game and how it fairs to the original are also listed in this document. The future scope of such an application is also discussed.**

Keywords—Gesture recognition; computer vision; OpenCV.

I. INTRODUCTION

In vision-based interfaces for video games, gestures are used as commands for the games instead of pressing buttons on a keyboard or moving a mouse. In these interfaces, unintentional movements and continuous gestures must be supported to provide the user with a more natural interface.

The paper I decided to work on is the classic snake game.
This is a game in which, the user uses a red coloured object and moves it around in front of a webcam so that a trail follows the object on its image and represents the shape of a snake. The size of this snake keeps on increasing in size and the game becomes progressively more difficult to play.

This project has been coded in Python . The libraries used are NumPy and OpenCV . This project is considered to be under the field of gesture recognition which is a field which is gaining widespread popularity as it makes it very easy for the user get some particular tasks done. Gesture recognition has slowly but surely created a stronghold in the gaming industry and now moving beyond it in every aspect. Together with computer vision it is enabling for a much better user experience and dynamic and user-friendly interfaces.

In our paper, we used techniques like colour detection contour detection drawing, placing png over image and checking for intersection of line segments to achieve results.

II. BACKGROUND

The background of the implementation lies in the idea to develop an interesting gaming application for the children in which they can get immersed and have a fabulous experience.

The aim was to deliver the following features to the user :

- Error free calculation of score
- Prompts to follow next instructions.
- Accurate object recognition to prevent false results.
- Negligible response time to react to object movement.
- Prompt readjustment of target location.
- Easy to exit application.

The game should be addictive and provide a strong will to succeed and stay in the game. Every object class has its own special features that help in classifying the object.

object recognition is that sub-domain of computer vision which helps in identifying objects in an image or video sequence. With more efficient algorithms, objects can even be recognized even when they are partially obstructed from the direct view. Various approaches to this task have been implemented in the past years.

III. HUMAN COMPUTER INTERACTION

HCI is the domain of computer science which deals with how users interact with their systems and how the user experience and the user interfaces can be optimized to improve user satisfaction.

The following points discuss this project with respect to the HCI paradigm.

The input channel used is the snake head which is an object that is recognized by the interface through Computer Vision and its relative position is mapped at every instance of the way. Movement of the said object causes a change in the relative motion of the snake drawn on top of the image captured through the webcam. Thus the interface is quick and responsive.

object recognition and color detection in computerized pictures has gotten one of the most significant applications for enterprises to ease client, spare time and to accomplish parallelism. This is certainly not

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

another method, yet improvement in object recognition is still required so as to accomplish the focus on target to be all the more productively and precise.

The principle point of considering and investigating computer vision is to recreate the conduct and way of human eyes legitimately by utilizing a computer and later on build up a framework that diminishes human effort.

Computer vision is such sort of research field which attempts to see and speaks to the 3D data for world items. Its fundamental reason for existing is remaking the visual parts of 3D protests in the wake of breaking down the 2D data extricated. Genuine 3D objects are spoken to by 2D pictures. The procedure of item discovery examination is to decide the number, area, size, position of the articles in the input image.

object detection is the essential idea for following and acknowledgment of items, which influences the proficiency and precision of object acknowledgment.

The normal object recognition technique is the color based methodology, distinguishing objects dependent on their color intensities. The technique is utilized on account of its solid flexibility and power, be that as it may, the recognition speed should be improved, in light of the fact that it requires testing every single imaginable window by thorough pursuit and has high computational unpredictability.

List of Nielsen heuristics applicable to the game

1. Visibility of system status : The score is updated in real time and user is always aware of his/her status.
2. Match between system and real world : The game communicates with the user in a language that the user will understand.
3. User Control and freedom : Supports undo and redo as it is has three states I.e start, playing and end.
4. Consistency and standards : The rules of the game are clearly defined and standardized for all plays.
5. Recognition rather than recall : Game is played based on recognition of target and reaction of user.
6. Flexibility and Efficiency of use : The interaction takes place at real time and is there is no concept of lag
7. Aesthetic and minimalist design.

Not applicable :

1. Help and Documentation.
2. Help users, recognize ,diagnose and recover from errors.
3. Error prevention.

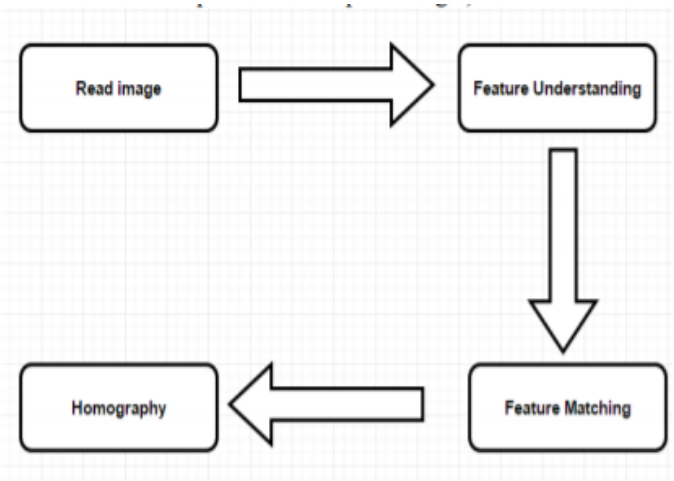


Figure 1 : Design of gesture recognition process

Design

1. Conversion of each frame of the video into hsv format from rgb format.
2. Setting of lower and upper limits of hsv values being tracked.
3. These lower values are then converted to Numpy array values which are processed by the computer.
4. Colour is located on the screen and their position coordinates are tracked.
5. The centre of the object is identified after using morphological transformation.
6. The object is tracked and a trail mask is created which follows the course of the red object and takes the form of the body of the snake.
7. The snake increases in size at the rate of 40 px per second and the game becomes progressively more difficult to play.
8. An apple is generated at random positions inside the frame. Whenever the snake passes through the apple, the score increases by 1.
9. Whenever the snake intersects with its own body, the game ends and “GAME OVER” and the game ends.
10. The user can then press any key to exit.

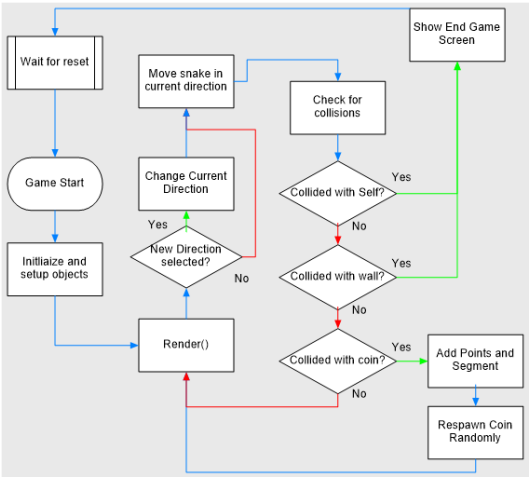


Figure 2 : Flowchart for Snake Game

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 11, November-2019

TECHNOLOGIES USED

Python

Python is an interpreted high level programming language used for a wide host of functionalities.

Libraries used :

1. OpenCv

OpenCV (open source computer vision) is a library of programming functions mainly aimed at real-time computer vision.

2. NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

ADVANTAGES

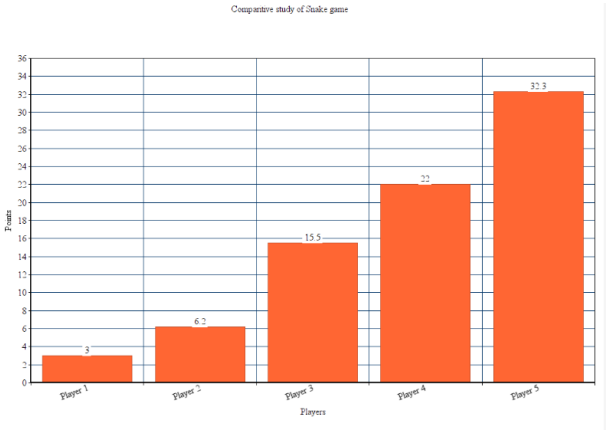
The conceivable outcomes of utilizing computer vision to take care of genuine issues are many. The essentials of item recognition alongside different methods for accomplishing it and its extension has been talked about. Python has been favored over MATLAB for coordinating with OpenCV on the grounds that when a Matlab program is run on a PC, it gets caught up with attempting to translate all that Matlab code as Matlab code is based on Java.

RESULTS

User Performance in the game . Average score in five plays.

Player	Avg Score
Player 1 (<10 yrs)	3
Player 2 (10 - 20 yrs)	6.2
Player 3 (>20 yrs)	15.5
Player 4 (>20 yrs)	22
Player 5 (>20 yrs)	32.3

Table 1 : User Performance



FUTURE SCOPE

Computer vision is as yet a creating discipline, it has not been developed to that level where it very well may be applied legitimately to genuine issues.

After few years"Computer vision and especially the item location would not be any progressively advanced and will be universal. For the present, we can consider object location as a sub-part of AI.

REFERENCES

[1] HOWSE, JOSEPH. *OPENCV: COMPUTER VISION PROJECTS WITH PYTHON*. PACKT PUBLISHING LIMITED, 2016

RESEARCH INTERNSHIP REPORT

User Journeys: Analysis, Visualization and Prediction

(NEXT Labs, Mphasis Limited, Bengaluru)

June 2019

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)}

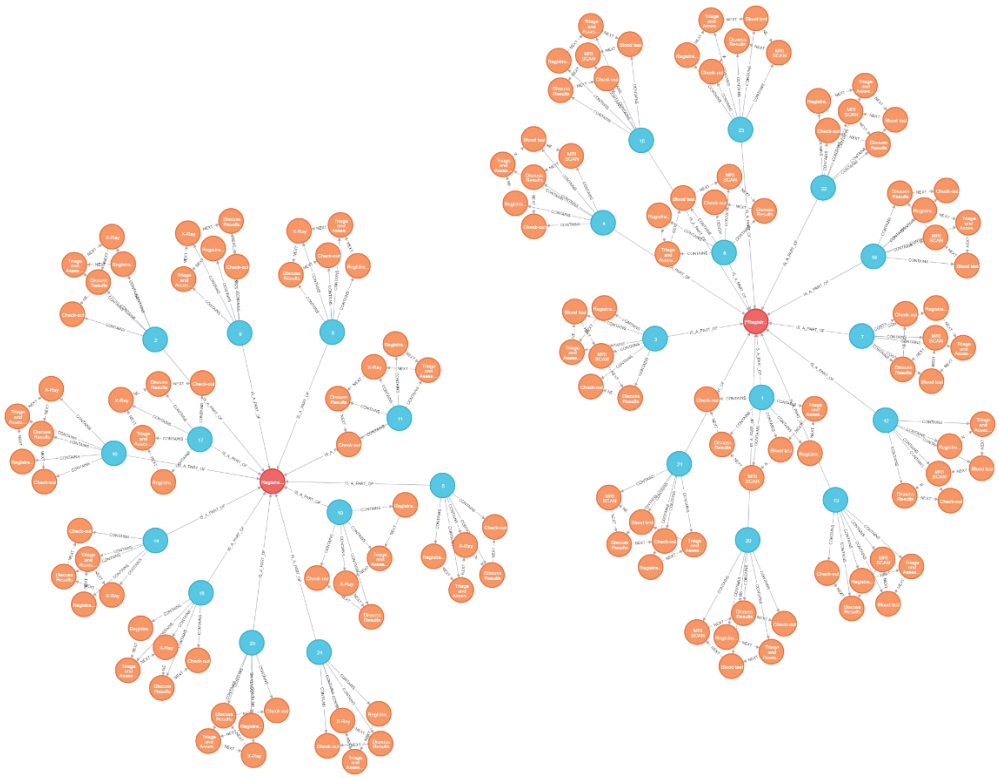
DATABASE STRUCTURE

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.

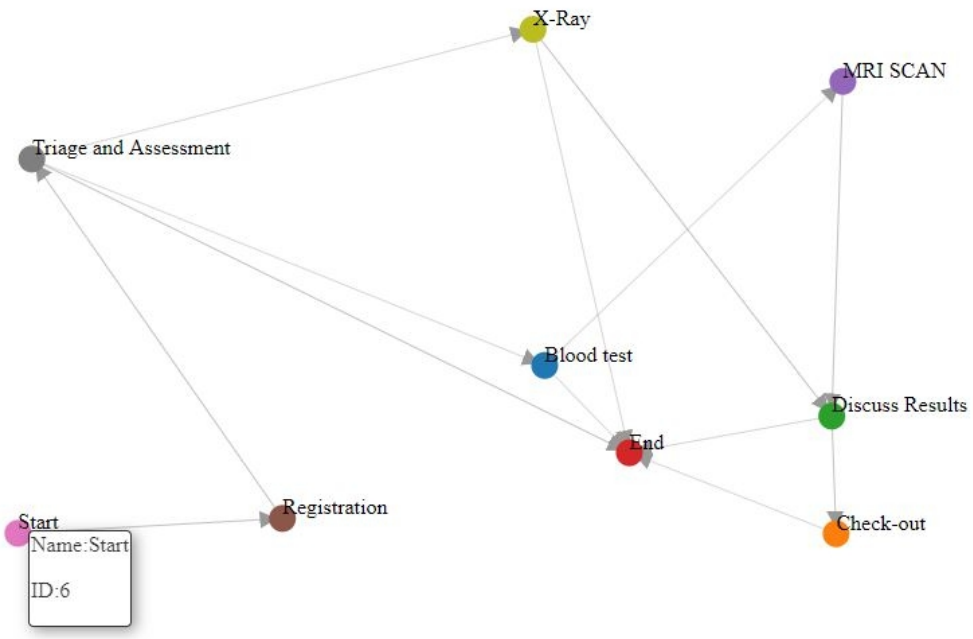
My first approach was to use the d3.js JavaScript Library to visualize the process map.

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.

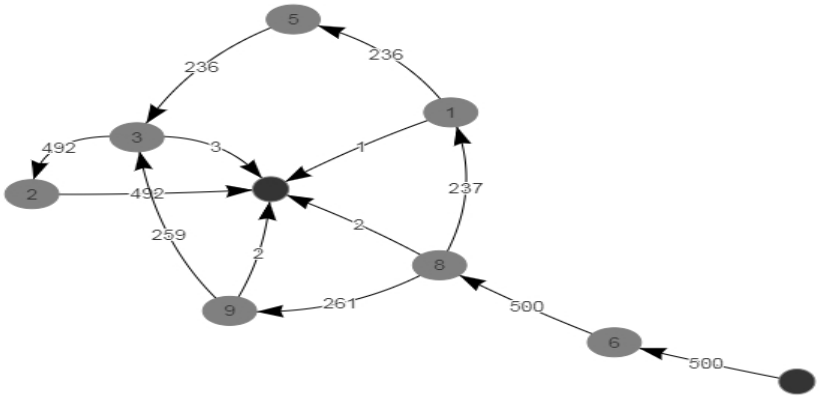
Objectives fulfilled with this approach:

- ☑ 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

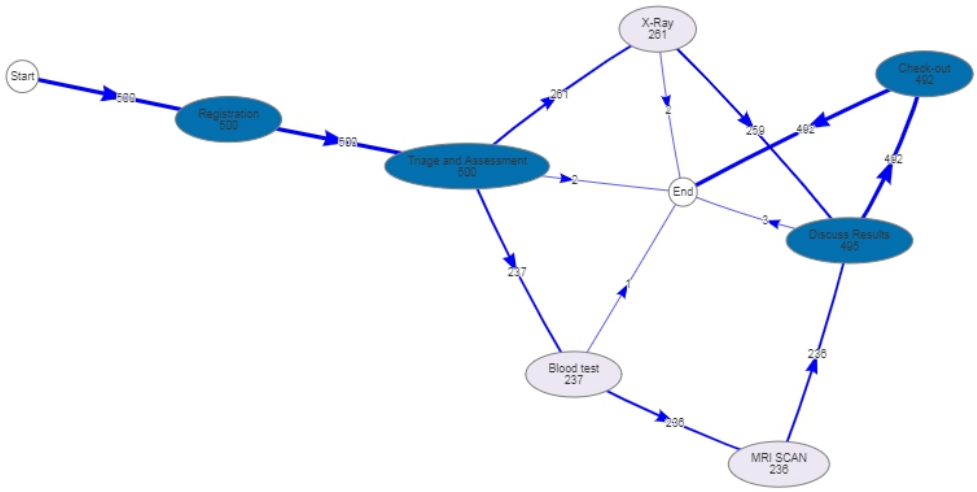
visNetwork is an R package for network visualization, using vis.js JavaScript library. It is based on HTML widgets and so it is compatible with R-Shiny. It proposes all the features available in vis.js API.

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.

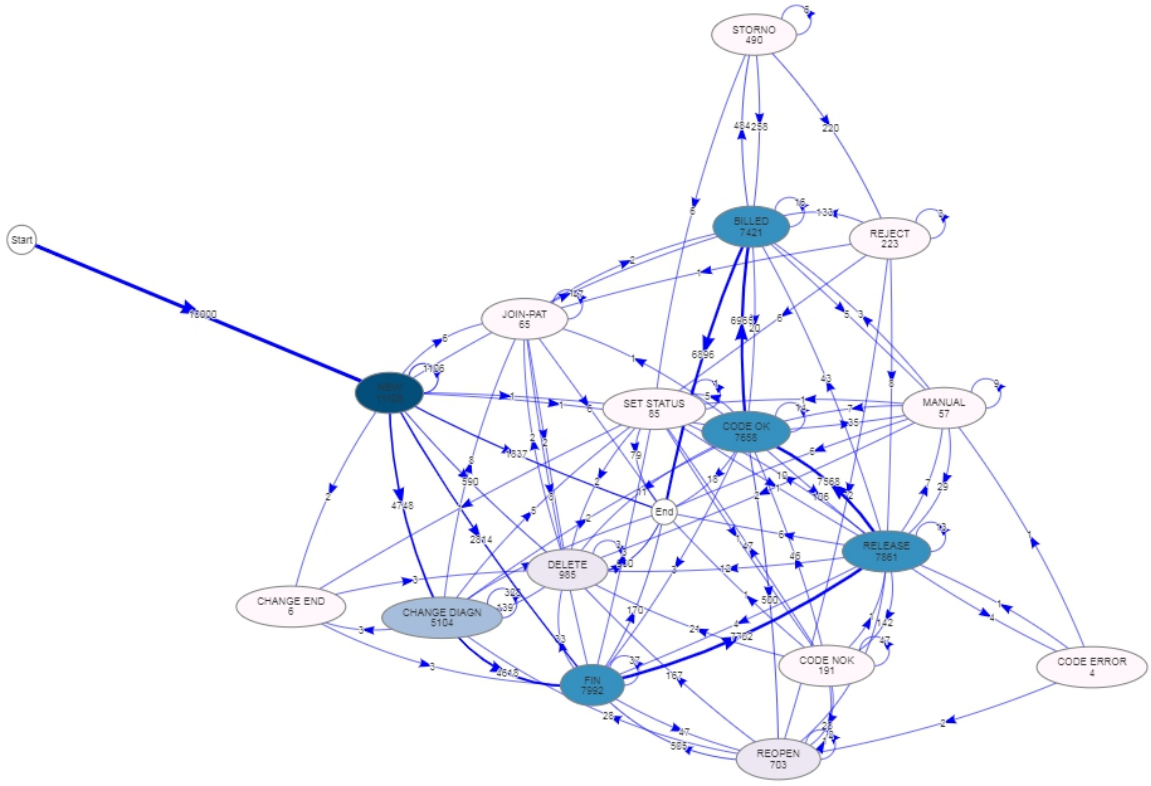
FUTURE ENHANCEMENTS:

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

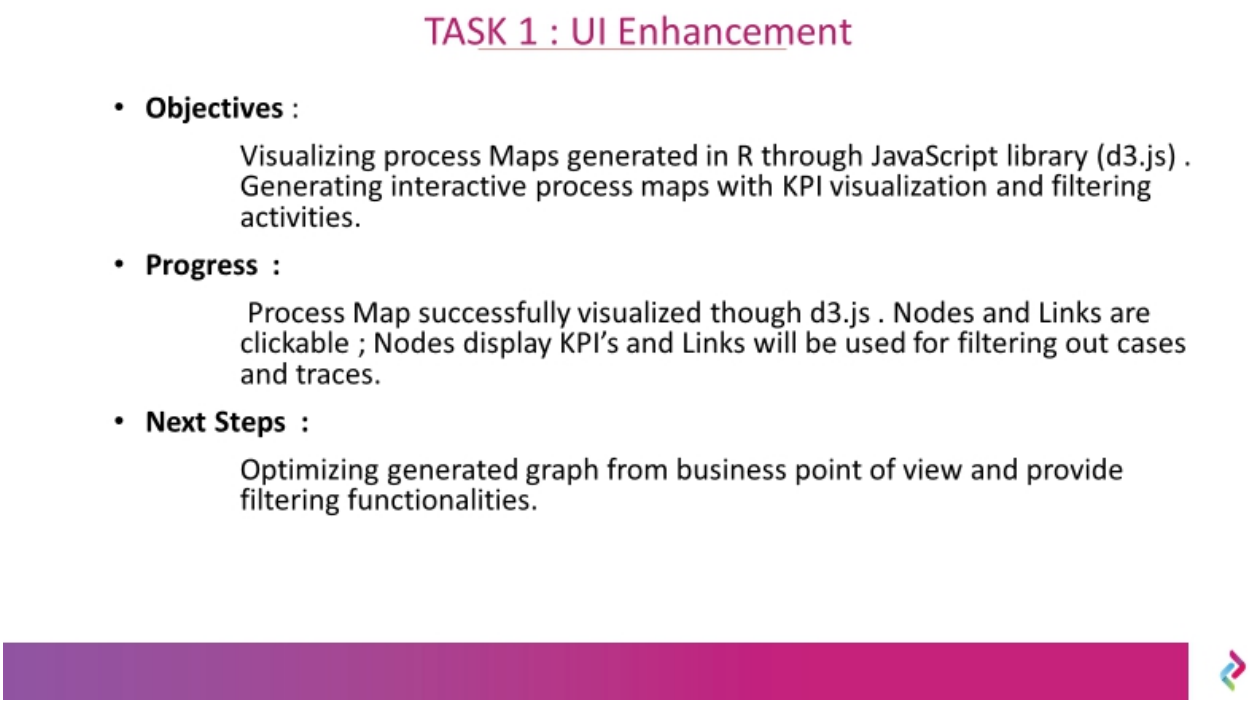
PRESENTATIONS GIVEN:

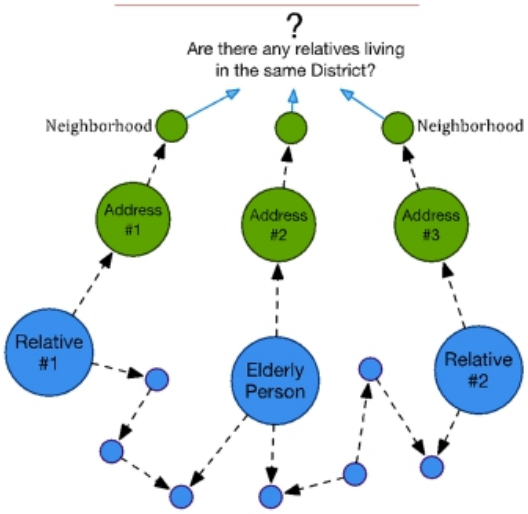
Presentation 1:

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

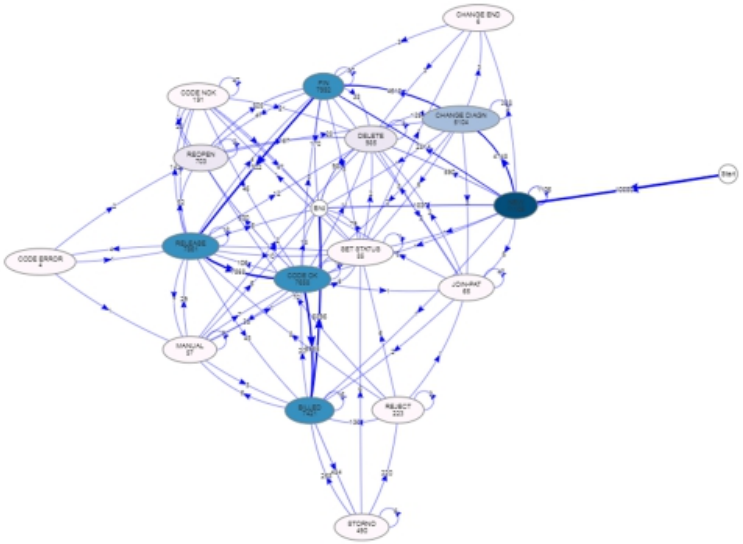
Audience : Mphasis, NEXT LABS Team.

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.

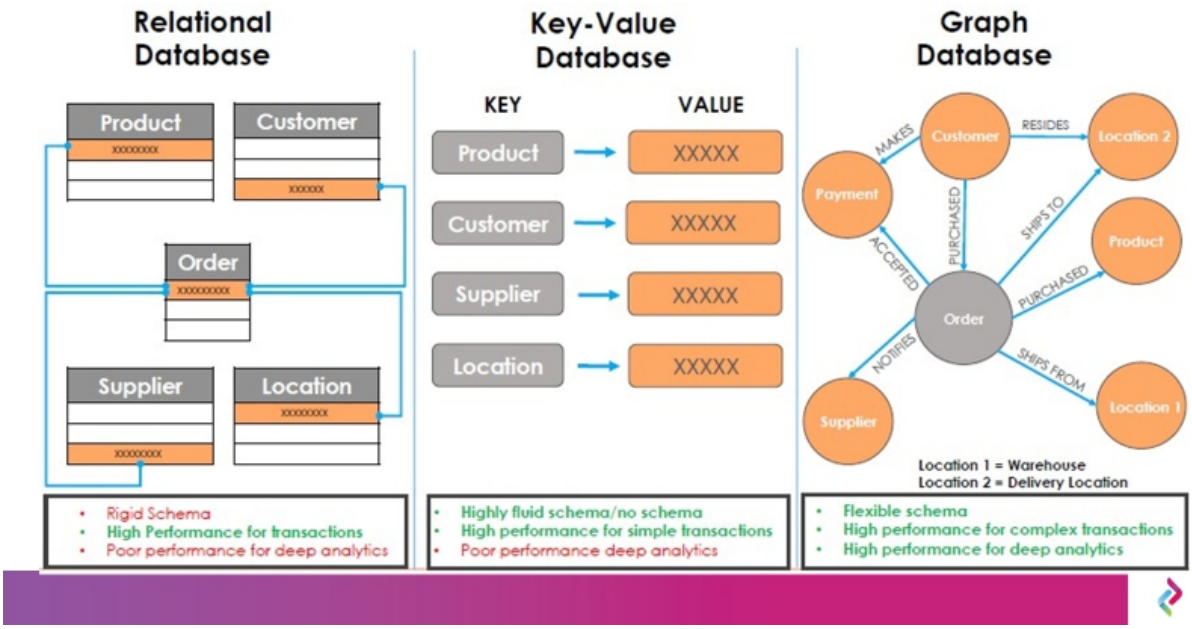
- **Challenge**: None of the network visualization packages had animation options which were similar to the objective.
- Ndtv3 and gganimate did show some animation capabilities, but were very limited.

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

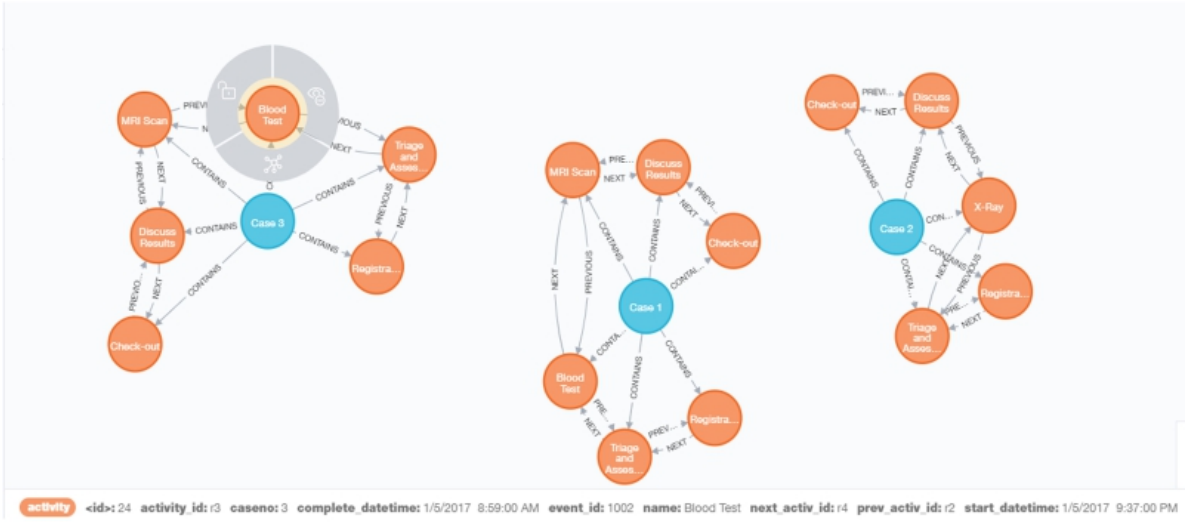
- **Approach2:** Researched on function vis.AnimateTraffic() – challenge – tokens travelling between nodes, do not follow the time constraints, merely run from source to destination. Need for inclusion of time and duration parameters.

- **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