MASTER OF TECHNOLOGY
(INTELLIGENT SYSTEMS)

PROJECT REPORT

Enterprise Knowledge Graph System
(Knowledge Graph Solution that leads to Enterprise AI)

GROUP MEMBERS

YANG LU YI

YIN TIAN SHI

YU YU

MASTER OF TECHNOLOGY Institute of System Science, National University of Singapore

Contents

1.0 Executive Summary

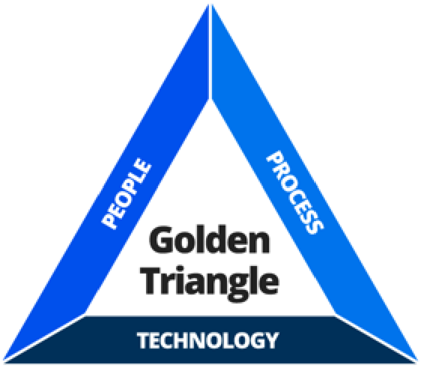
It is well-known that web contains massive amount of knowledge, in the form of structured and unstructured data. Extracting, re-organizing and integrating these fragmented knowledges into a large-scale knowledge graph will be the development trend of the Internet. Currently, the construction and application of knowledge graphs have achieved great success, however, they have not been full used in many industry fields, especially for enterprise domain. On the contrast, there is a huge demand from big companies and different SME. Furthermore, all our team members are working in IT industries. Hence, we built a web-based Enterprise Knowledge Graph System, which is targeted to internal stakeholders (e.g. Enterprise Architect, Solutions Designer and Business Analyst etc.)

The system provides comprehensive graph view and necessary text summary of business operation process regarding to "360-degree Scan", "Relationship Scan", "End-to-End Process Scan" portions. Its "Recommend Solution" function gives users with the specific recommendation about optimized operation process and solution design strategy. While interact with the intelligent system, uses are able to input free text or select listed options to understand whole business structure. Through adjusting the values of resource count and load distribution for nodes "People", "Process" and "Technology" on "Resource Simulation" page, to get latest Cost and Time properties for requests "Request-to-Answer process with lowest cost or shortest response time".

The biggest highlights of our system are the mature Framework (named FRAMWORX), Cognitive User Interface design (supported by Google Dialogflow) and solid basics with 3500+ nodes 3300+ relations in Graph database. In addition, with adoption of the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology and other algorithms (e.g. decision tree, knowledge graph, BFS, DFS and Dijkstra etc.), the overall system is definitely stable and efficient to respond correct results in various forms including graphs, texts and spreadsheets.

2.0 Business Problem Background

To maximize profits, majority of enterprises always try to continuously develop end-to-end business operation process and solution architecture design in both high quality and efficiency. And building a good operating model is the necessary foundation, the most common operating model is PPT (People, Process and Technology). It's about People (e.g. internal staffs, external customers and suppliers) who participate in the operation process to find out valuable solution strategy with adoption of Technologies (e.g. IT infrastructure, applications and equipment etc.)



That is, how to balance these three elements and their relationships well has become an extremely significant factor for enterprises. Furthermore, it's really difficult to maintain the large masses of data among People, Process and Technology, as the data is not static, turnover of staffs, frequent changes in the process and complexity of technologies could be potential concerns occur at any time. Employers have to spend much time, cost and resources to ensure real-time data update correctly.

In general, there is a wide range of data sources, data types for an organization or a typical department. For instance, massive structured & un-structured data can come from interior or exterior of the company, 3rd-party or website. However, the traditional way of data management is not effective, which requires stakeholders to manually access the fixed system/ SharePoint/ Public Folders to get information. Sometimes, it even relies on brain storage.

In order to make use of the data (PPT elements and relationships) in a good way, we referred to the new type of massive knowledge management and service model based on the "Semantic Network", which was introduced by Google in 2012, called Knowledge Graph. After many years of development, Knowledge Graph has been applied to different industries of Artificial Intelligence. In particular, the enterprise-based Knowledge Graph can support to build data models from multiple dimensions, conduct a comprehensive risk assessment, then provide systematic and reasonable recommendations for internal stakeholders.

3.0 Market Research

Knowledge Graph can describe concepts, entities and mutual relationship from objective world in a structured form, then represent friendly too much internet info to human beings in an understandable and acceptable way. The technology of Knowledge Graph incorporates cognitive computing, knowledge discovery, representation and reasoning, semantic web, data mining, machine learning and natural language processing etc. It is a significant branch of knowledge engineering in artificial intelligence field.

From different applications areas, Knowledge Graph are divided into two parts, general and domain-specific Knowledge Graphs. The General Knowledge Graph can be visually regarded as a "structured encyclopaedia for general domain", which contains lots of wide common-sense knowledge. In addition, domain-specific Knowledge Graph is also called industry/ vertical Knowledge Graph, it can be regarded as an "industry knowledge above on semantic technology".

Since our team members are all working in IT industries, as Designer, Developer and Solution Engineer respectively, we are more interested in how to build an intelligent business system with the adoption of Knowledge Graph, which is targeted for internal stakeholders like us. In order to design a valuable business system, we investigated some current intelligent systems that applied in different commercial domains.

Listed the typical business systems for reference.

1) TKG (Tencent Knowledge Graph)

A one-stop platform integrating graph database, calculation engine and visualization analysis. It supports extraction and fusion of heterogeneous data, the storage and calculation of huge node and relationships, graph data mining algorithms and rich visualization solutions etc.

Application Scenarios	Feathers
• *Finance/Banking domain:* For Corporate Business: Store big data of customers. Sort out customer's relationships to develop potential business value. For Private Business: Predict the potential risk of new customers, detect malicious user groups. • *Safety and Security domain:* Help security analysts to discover potential risks, support public safety agencies to conduct intelligence analysis, criminal gang tracking and early warning. • *IoT domain:* Provide a full set of solutions from engine-level products to the landing of industry knowledge, help customers mine hidden huge value from data.	• *Convenient knowledge graph construction:* Allow users to upload raw data, define a knowledge model and configure the mapping between of them. Users can also upload text data, the system can automatically extract characters and relationships to build a knowledge graph. • *Efficient graph for online query* • *Rich graph calculation models* E.g. graph embedding, clustering and community discovery etc. • *Various visual analysis methods:* Support node multi-hop expansion, path search analysis, mixed condition query, calculation result display and other operations. • *Independent deployment:* Can be completely privatized and customized, operate well in an independent, private and secure environment.

2) E-Profile (Enterprise Profile)

It provides multi-dimensional information query for enterprises, mutual relationships between employers and employers, or employers and employee, then generate potential values and risks of customers.

Application Scenarios	Feathers
• *Business Evaluation:* Provide audits for enterprise applications and settlements, evaluate the risk and growth indexes for suppliers, customers & distributors, identify bid rigging during supply chain and loan process for upstream and downstream enterprises. • *Product Recommendation domain:* Explore potential business opportunities for enterprises, recommend products according to customer characteristics and behavior and rank business opportunities and customers etc. • *Smart City domain:* Analyse the investment and financing for regional enterprises.	• *Holographic portrait:* Weekly update the information query for 100 million of SME and individual households, provide 360-degree information query, e.g. investment, legal representative, corporate annual report, recruitment and risk etc. • *Relationship network:* Provide the equity relationship inquires for individual and multiple enterprises, quantitatively evaluate the relationship between them. • *Risk monitoring:* Monitor the risk dynamics for enterprises in real time, such as business changes, abnormal operations and justice, explore the risks of affiliated companies and natural persons based on the relationship network.

Overall, many internet companies have realized the strategic significance of Knowledge Graph and have invested the layout of knowledge maps. At the same time, we also find that the techniques of knowledge graph are still in early stage of development, most business knowledge graphs just can be applied in limited scenarios as we understood that majority of business knowledge graph systems are just applied in finance and healthcare domains, and more human intervention is still required in the process of constructing knowledge graph to ensure its accuracy. Hence, the whole constructing steps (knowledge types and representations, knowledge acquisition, knowledge fusion and application) of knowledge graph need to be further advanced by the academia and industries. It can be seen that future development of knowledge graph will be more characterized and intelligent.

4.0 Project Objectives & Success Measurements

Our team aims to create an Enterprise Knowledge Graph system that can show up systematic representation for whole enterprise architectures, allow internal users to perform searching and reasoning requests between nodes (people, process, technology) and mutual relationships, also provide valuable recommendation for users' reference with the optimized operation process and solution design strategy in customer interaction management portion. As the project is focused on business operation process, it would be more useful for internal stakeholders in a company, such as Enterprise Architect, Solution Designer and Business Analyst, or anyone who is interested in utilizing AI technology to assist in the analysis and reasoning work in the enterprise environment.

In order to make sure stability and efficiency of the system, the overall design is based on the mature framework "Framworx", and different algorithms have been adopted well in function achievements. For instance, decision tree, knowledge graph, BFS, DFS and Dijkstra algorithms are used for searching and reasoning requests, cognitive techniques are integrated into the rule engine to capture users' text input and then translate into searching and reasoning rules.

While interact with the accessible Web intelligent system, users can observe the text brief summary and corresponding knowledge graph after selecting suitable phrases or inputting first letters of phrases on search bar. Furthermore, extra spreadsheets will be listed if users want to get recommended solutions about "Request-to-Answer with lowest cost" and "Request-to-Answer with shortest response time".

Use cases for Searching and Reasoning requests are as below:

Use Case	Functionality	User Input	System Output

Searching Request:
The related nodes and mutual relationships can be used for impact analysis and risk assessment.
	Search bar is targeting for user to key in search content in a structured text format (close to natural language).
	With application of Cognitive System techniques, the system is designed to understand and map the text into searching rules, fulfil user's searching request by
		"360 Scan for People <Sales Agent> in <Market Sales Domain>";
		"Relationship scan between Process <Customer Interaction Management> and Technology <Customer Self-Management>";
			A 360-degree overview graph of both the node and relationships can be presented on the web page;
			Brief text summary is also generated for reference.

	using Machine Reasoning technologies. The search bar supports both error correction and suggested features for the original text.	"End to End Process Stream Scan for <Request-to-Answer>".	
Reasoning Request: Recommend the optimal process stream according to user's specific requirement. E.g. Lowest cost to fulfil Request-to-Answer request from customer & Shortest response time to fulfil Request-to-Answer request from customer.	Same as above.	"Recommend a solution to fulfil Request-to-Answer request with lowest cost"; "Recommend a solution to fulfil Request-to-Answer request with shortest time".	A 360-degree overview graph with necessary text summary represents the optimal end-to-end business operation process stream among People (customer, customer service representative, etc), Process (Customer Interaction Management, etc) and Technology (Customer Self-Management application, etc).
Simulation of dynamic Transactional data and observe change in the Recommend a solution to fulfil Request-to-Answer use case.	System shall allow user to simulate the dynamic transactional data, such as Load, Resource count and Load Distribution for each node in the Request-to-Answer process, and the system shall be able to spread and distribute the load along the path properly. And if user perform the Recommend a solution to fulfil Request-to-Answer use case again, the result shall be changed according to new search result.	Load count to the whole process. Load Distribution property for each node, for system to distribute the load according to the rate, the higher the value, the more load will be distributed to a node. Resource Count for each node, the higher the value, the more resource will be added to this node to balance the Load.	An updated Graph path with new properties in each node. And new search result for Recommend a solution to fulfil Request-to-Answer use case.

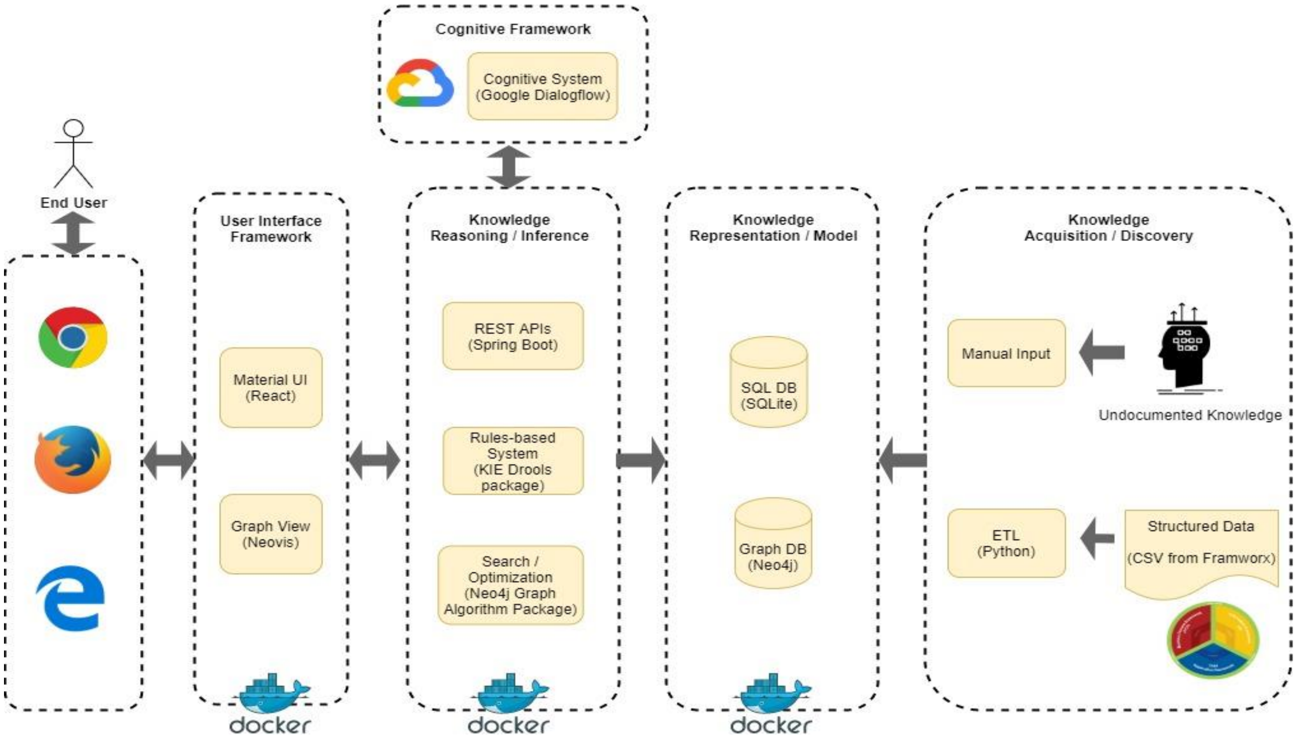
In addition, those following success measurements have been considered for our project.

- System is able to identify user's inputs (keywords/ Phrases) which are relevant to "360-degree Scan/ Relationship Scan/ End-to-End Process Scan/ Recommend Solution".
- System is able to fulfil "360-degree Scan/ Relationship Scan/ End-to-End Process Scan" use cases successfully less than 30 seconds.
- System is able to fulfil "Recommend Solution for Request-to-Answer" use case successfully less than 1 minute.
- System can be deployed to any Operating System that supports Docker container.
- Documentations like project report, user guide and video have done.
- Customers (Enterprise stakeholders like Yang Lu Yi, Yin Tian Shi, Yu Yu) and Lecturer accept the system deliverables.

5.0 Project Solution (To detail domain modelling & system design.)

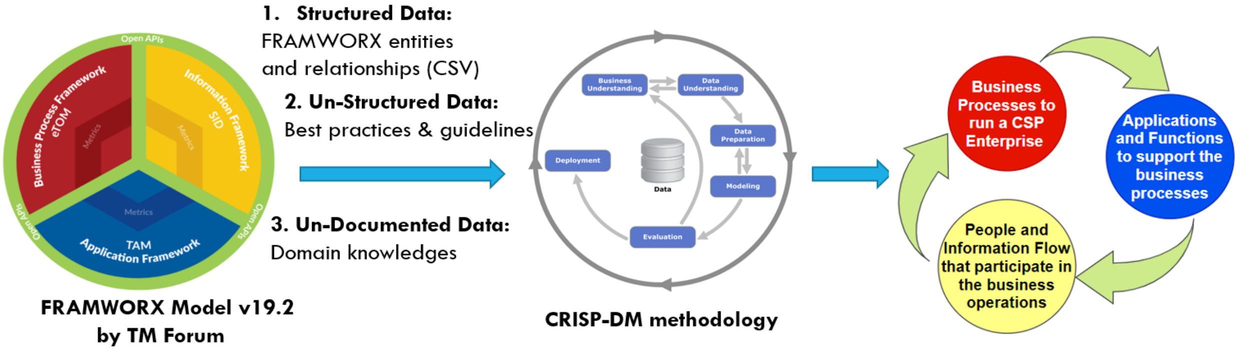
We built a web-based system for end user to communicate with the intelligent system through near to natural text input. Firstly, the Cognitive User Interface design is supported by Google Dialogflow, rendered in the Web Brower by React framework in a Material style.

Neovis software library helps represent the Knowledge Graph in UI for all use cases. Secondly, the Knowledge Reasoning module integrates Java Spring Boot Framework, KIE Drools Rule Engine and Neo4j Graph Algorithm packages. Thirdly, both Neo4j Graph Database (stores Knowledge Graph model) and SQLite Relational Database are solid basics for Knowledge Representation. And all structured and unstructured data are processed on the Knowledge Acquisition/Discovery stage. Finally, all modules are contained in Docker container.


Cognitive Framework
Cognitive System (Google Dialogflow)
End User
User Interface Framework
Material UI (React)
Graph View (Neovis)
Knowledge Reasoning / Inference
REST APIs (Spring Boot)
Rules-based System (KIE Drools package)
Search / Optimization (Neo4j Graph Algorithm Package)
Knowledge Representation / Model
SQL DB (SQLite)
Graph DB (Neo4j)
Knowledge Acquisition / Discovery
Manual Input
Undocumented Knowledge
ETL (Python)
Structured Data (CSV from Framworx)
docker
docker
docker

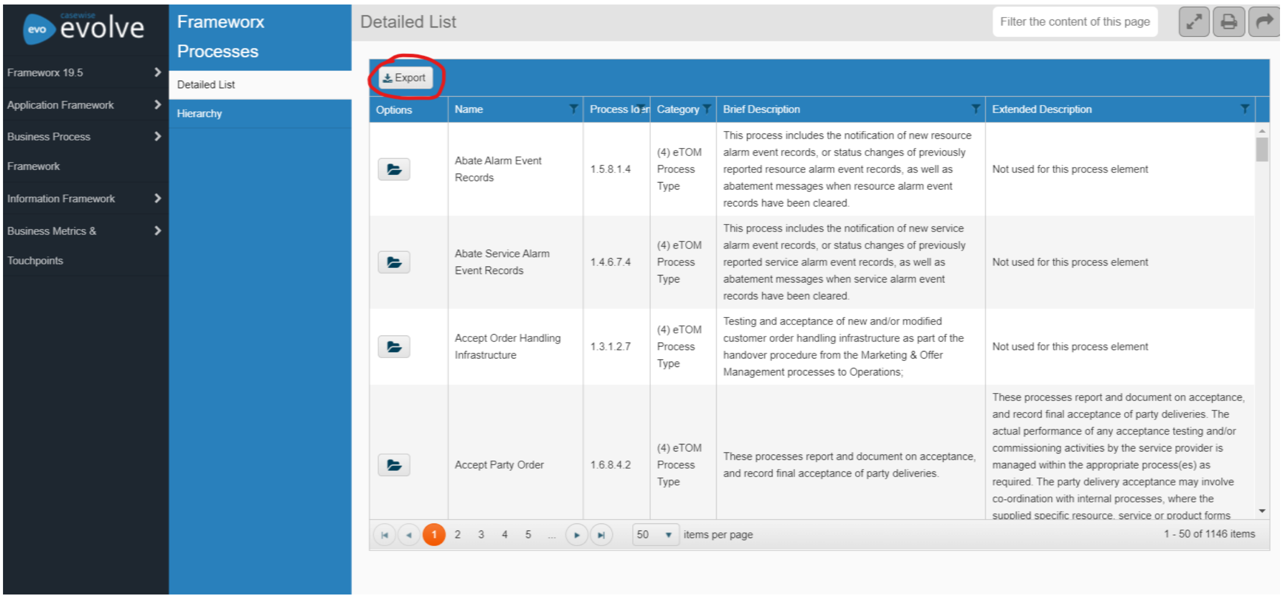
5.1 Knowledge Discovery/Extraction

The purpose of knowledge discovery is to extract useful data from reliable sources and establish the basic foundation of overall information/ knowledge framework. We used FRAMWORX as the main source of knowledge, which is the industry-agreed business operation model for Communication Service Provider organization. And adapted the CRISP-DM methodology to extract structured and un-structured data from FRAMWORX. In addition, we also learnt typical CSP business operation processes regarding to Applications, People and information etc.


1. Structured Data:
FRAMWORX entities and relationships (CSV)
2. Un-Structured Data:
Best practices & guidelines
3. Un-Documented Data:
Domain knowledges
Open APIs
Business Process Framework eTOM
Information Framework SID
TAM Application Framework
Metrics
FRAMWORX Model v19.2 by TM Forum
Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Data
CRISP-DM methodology
Business Processes to run a CSP Enterprise
Applications and Functions to support the business processes
People and Information Flow that participate in the business operations

5.1.1 Structured Data from TM Forum FRAMWORX

The data is browsable in web browser and files can be exported in CSV format. (URL: http://casewise.tmforum.org/evolve/statics/frameworx/index.html#cwtype=index&cwview=home)

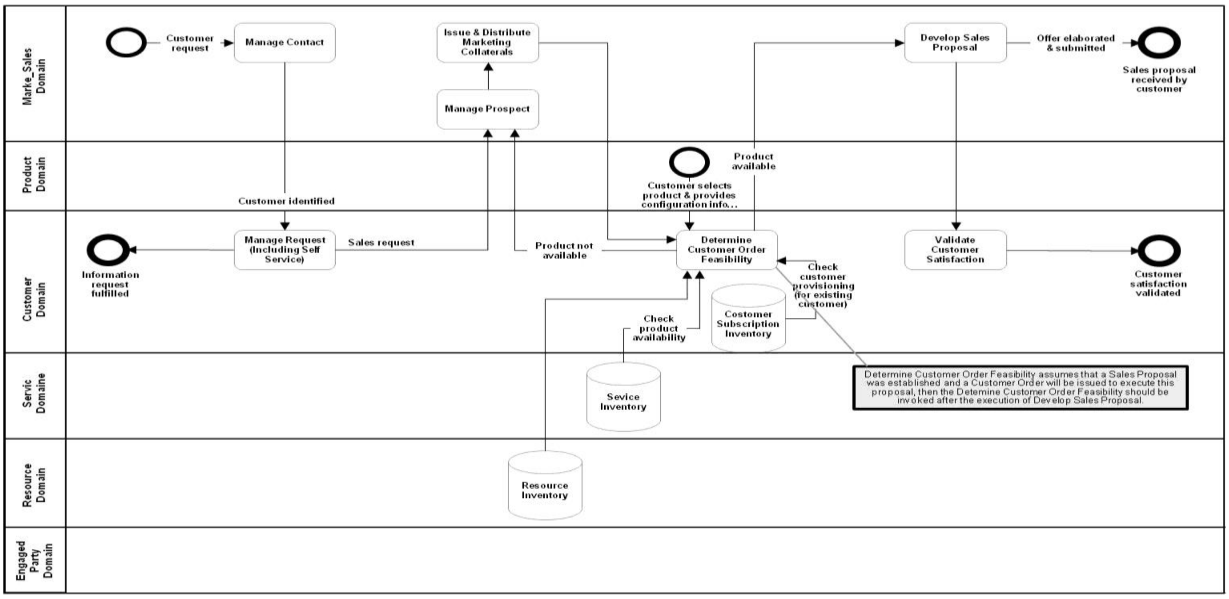


Options	Name	Process Ident	Category	Brief Description	Extended Description
📂	Abate Alarm Event Records	1.5.8.1.4	(4) eTOM Process Type	This process includes the notification of new resource alarm event records, or status changes of previously reported resource alarm event records, as well as abatement messages when resource alarm event records have been cleared.	Not used for this process element
📂	Abate Service Alarm Event Records	1.4.6.7.4	(4) eTOM Process Type	This process includes the notification of new service alarm event records, or status changes of previously reported service alarm event records, as well as abatement messages when service alarm event records have been cleared.	Not used for this process element
📂	Accept Order Handling Infrastructure	1.3.1.2.7	(4) eTOM Process Type	Testing and acceptance of new and/or modified customer order handling infrastructure as part of the handover procedure from the Marketing & Offer Management processes to Operations;	Not used for this process element
📂	Accept Party Order	1.6.8.4.2	(4) eTOM Process Type	These processes report and document on acceptance, and record final acceptance of party deliveries.	These processes report and document on acceptance, and record final acceptance of party deliveries. The actual performance of any acceptance testing and/or commissioning activities by the service provider is managed within the appropriate process(es) as required. The party delivery acceptance may involve co-ordination with internal processes, where the supplied specific resource, service or product forms

1 2 3 4 5 ... 50 items per page 1 - 50 of 1146 items

5.1.2 Un-structured Data from TM Forum FRAMWORX

There are several documents about best-practice and guideline for applying the FRAMWORX model to business operation, which is available in TM Forum resource center (URL:https://www.tmforum.org/resources/), please refer to below example about the proposed business process flow stream.


Marke_Sales Domain
Product Domain
Customer Domain
Servic Domaine
Resource Domain
Engaged Party Domain
Customer request
Manage Contact
Issue & Distribute Marketing Collaterals
Manage Prospect
Develop Sales Proposal
Offer elaborated & submitted
Sales proposal received by customer
Product available
Customer selects product & provides configuration info...
Customer identified
Manage Request (Including Self Service)
Sales request
Information request fulfilled
Product not available
Determine Customer Order Feasibility
Check customer provisioning (for existing customer)
Costomer Subscription Inventory
Check product availability
Validate Customer Satisfaction
Customer satisfaction validated
Sevice Inventory
Resource Inventory

Determine Customer Order Feasibility assumes that a Sales Proposal was established and a Customer Order will be issued to execute this proposal, then the Detemine Customer Order Feasibility should be invoked after the execution of Develop Sales Proposal.

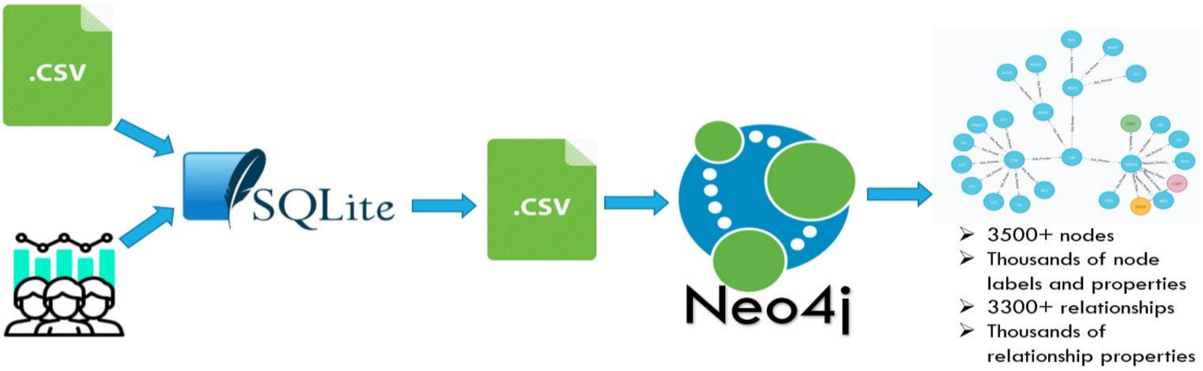
5.1.3 Domain Knowledge from our team members

We have consolidated our domain knowledge into the FRAMWORX model with additional information, such as cost, time and resource properties etc., which are required for certain

Business Process, Applications or People entities in the model. These properties will be useful in our knowledge representation and inference functionalities.

5.2 Knowledge Modelling/Representation

We consolidated our knowledge model in SQLite database, and created a script to extract and upload the model into Neo4j Graph database. The database involves 3500+ nodes and 3300+ relationships, nodes represent massive enterprise entities with separate labels and properties while relationships represent the structure and interactions with various properties. Each of the element plays an important role in Knowledge Inference module.



- 3500+ nodes
- Thousands of node labels and properties
- 3300+ relationships
- Thousands of relationship properties

1) Load the FRAMWORX data model into SQLite database

   We created 6 tables for the data modelling, the first 4 tables are eTOM, Function, TAM, SID, which are used for storing the raw data from FRAMWORX structured data source, the table schema is almost same as the CSV files from FRAMWORX.

2) Use Python script and manual input to transform the data into the graph representation

   The other two tables GRAPH_NODES and GRAPH_RELATION are used for representing the FRAMWORX data model in a graph form, where we can specify the node properties, relationships and its properties. These two tables are referred to below table schema (SQLiteStudio v3.2.1).

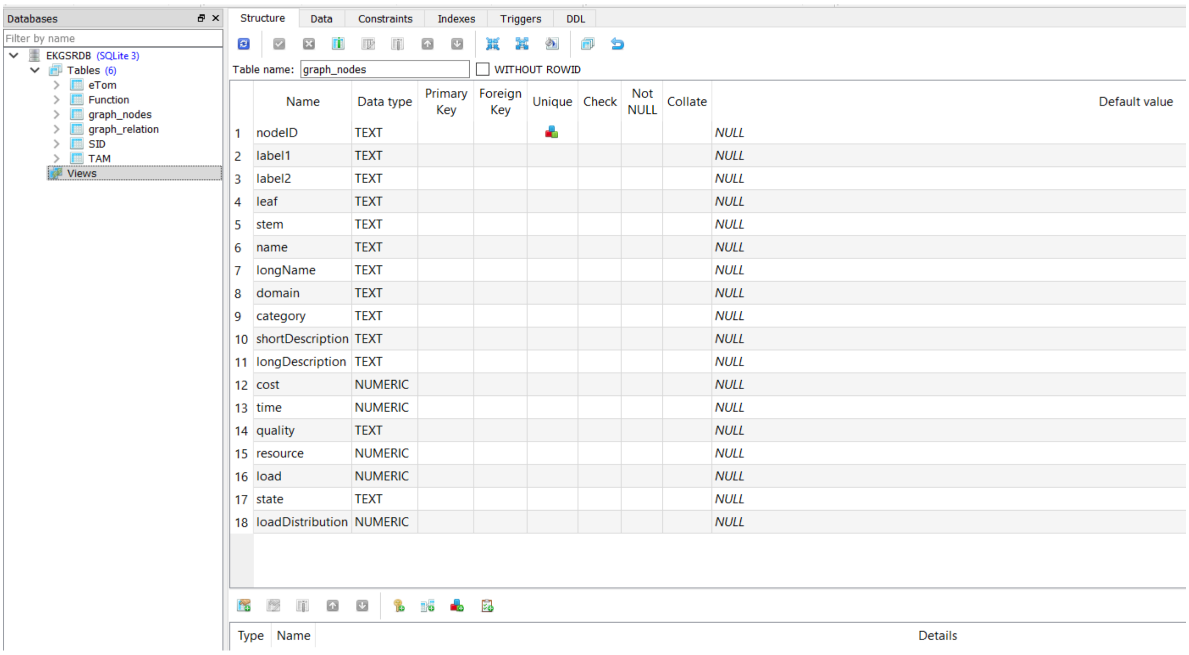


Table name: graph_nodes ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	nodeID	TEXT			■				NULL
2	label1	TEXT							NULL
3	label2	TEXT							NULL
4	leaf	TEXT							NULL
5	stem	TEXT							NULL
6	name	TEXT							NULL
7	longName	TEXT							NULL
8	domain	TEXT							NULL
9	category	TEXT							NULL
10	shortDescription	TEXT							NULL
11	longDescription	TEXT							NULL
12	cost	NUMERIC							NULL
13	time	NUMERIC							NULL
14	quality	TEXT							NULL
15	resource	NUMERIC							NULL
16	load	NUMERIC							NULL
17	state	TEXT							NULL
18	loadDistribution	NUMERIC							NULL

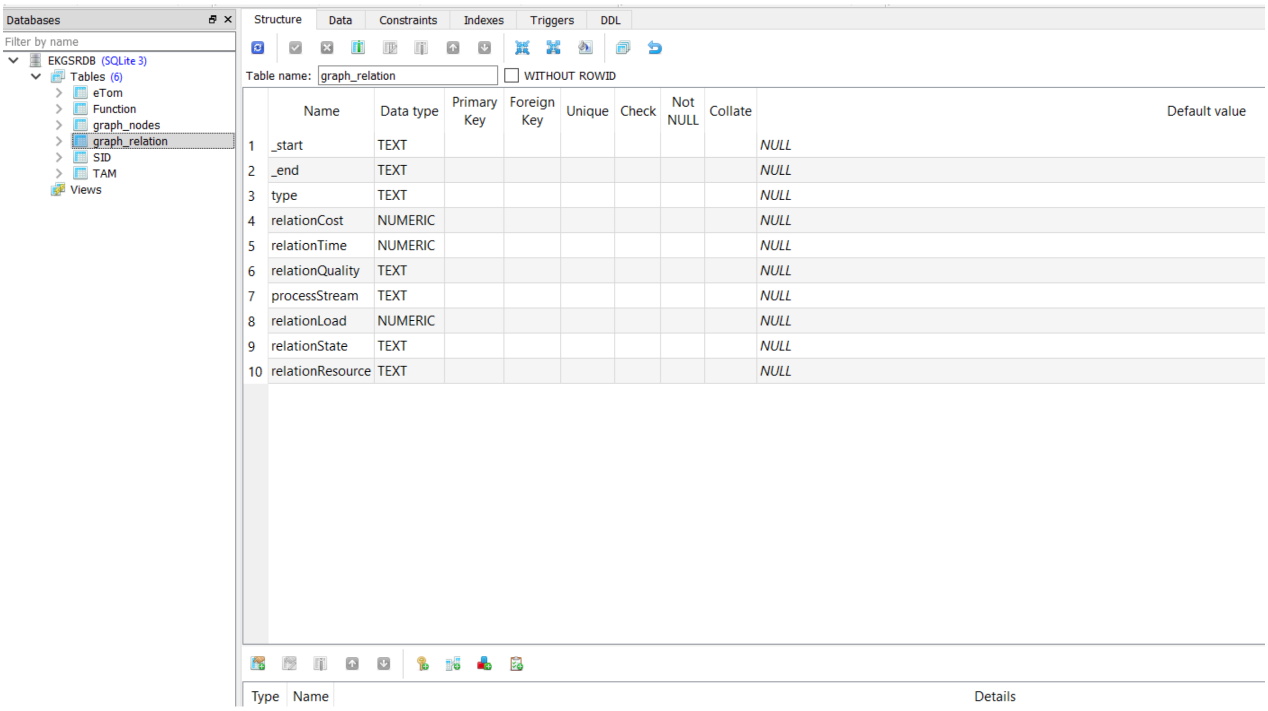
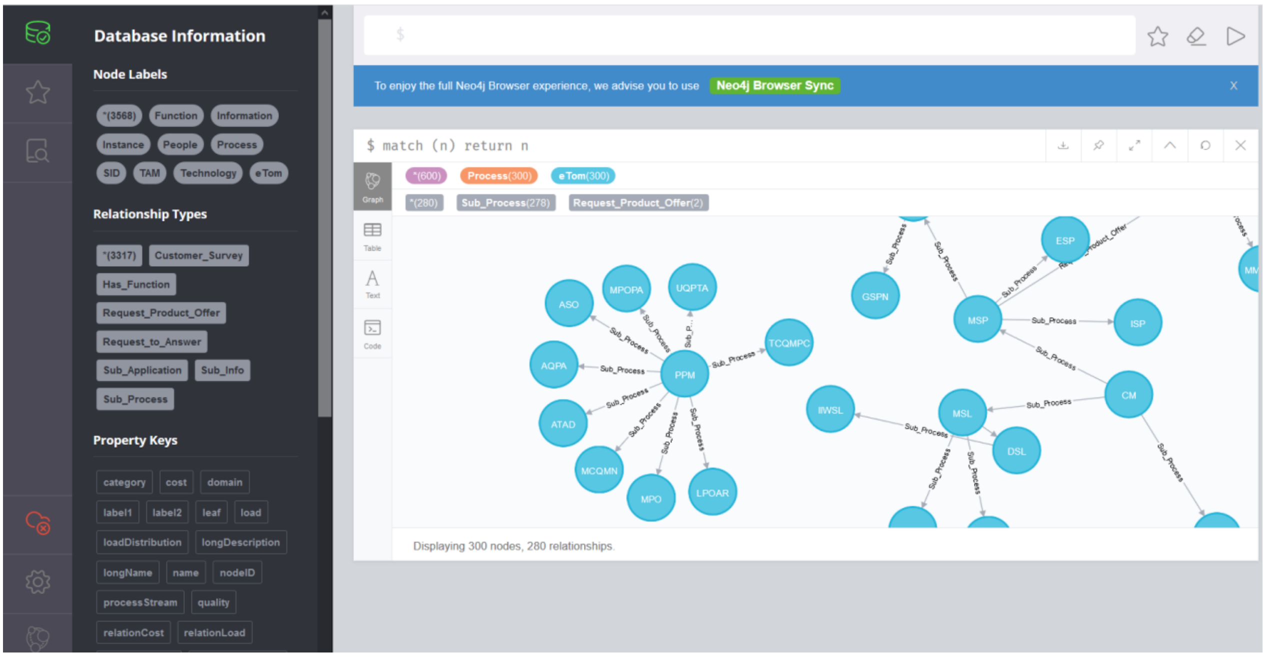


Table name: graph_relation ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	_start	TEXT							NULL
2	_end	TEXT							NULL
3	type	TEXT							NULL
4	relationCost	NUMERIC							NULL
5	relationTime	NUMERIC							NULL
6	relationQuality	TEXT							NULL
7	processStream	TEXT							NULL
8	relationLoad	NUMERIC							NULL
9	relationState	TEXT							NULL
10	relationResource	TEXT							NULL

3) Use Cypher script to transform the data into real Graph model in Neo4j Graph database.



To enjoy the full Neo4j Browser experience, we advise you to use Neo4j Browser Sync

```
$ match (n) return n
```

Displaying 300 nodes, 280 relationships.

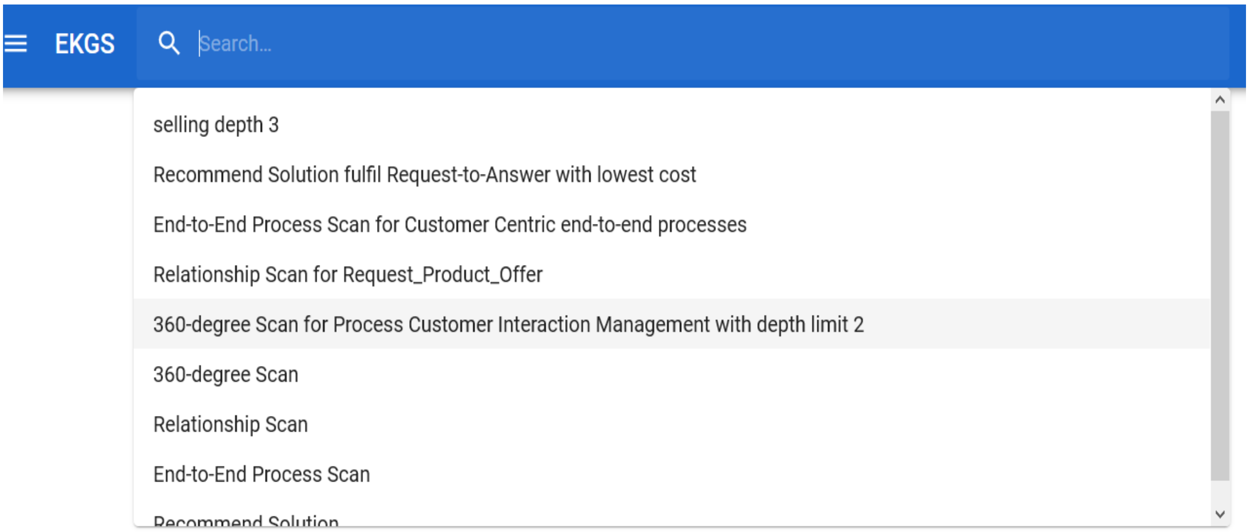
5.3 Knowledge Inference/Reasoning

After Knowledge model construction, we designed use cases by utilizing AI algorithms, graph query and traverse techniques. Please refer to 5.3.2 – 5.3.4 parts for details.

In addition to static data, the system also allows user to simulate the dynamic volume/load to the business process. Once given the no. of load, system can spread and distribute the load to all nodes and relationships along the path, and relevant cost and time of the path will be calculated accordingly, thus recommendation result can be updated based on different loads.

5.3.1 Rule-based Cognitive UI

We used Google Dialogflow to capture the user intent, Drools rule engine and decision tree techniques to map user intent to business rules that defined in the system.



selling depth 3
Recommend Solution fulfil Request-to-Answer with lowest cost
End-to-End Process Scan for Customer Centric end-to-end processes
Relationship Scan for Request_Product_Offer
360-degree Scan for Process Customer Interaction Management with depth limit 2
360-degree Scan
Relationship Scan
End-to-End Process Scan
Recommend Solution

If user clicks the search bar, system will automatically prompt the 4 default requests (360-degree Scan, Relationship Scan, End-to-End Process scan and Recommend Solution). If user selects any of the options, system will automatically prompt the search parameters available for that option and guide user to continue, such feather can prevent user from keying invalid command.

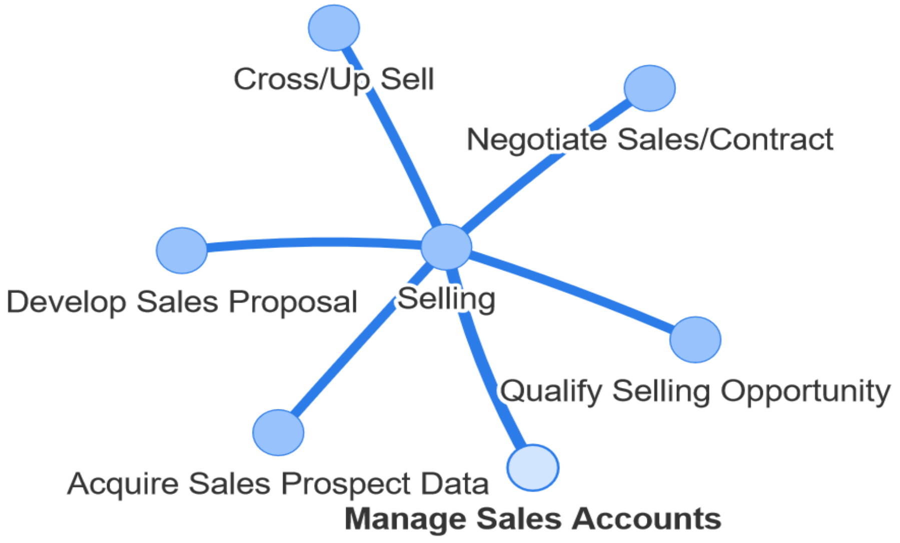
If user input is a free text or a question "what is xx" (e.g. what is selling/ what is customer interaction management), the system will try to match the free text or question to a node name in the graph database and trigger 360-degree scan rule. The default depth limit number of the graph is 3.

If user input is a free text + depth limit number X, the system will apply the depth limit X in the 360-degree scan rule parameter and return the result accordingly.

If user input is "process" + a process name, like "Request to Answer", then system will automatically match it to Recommend solution for Request to Answer process and trigger the search.

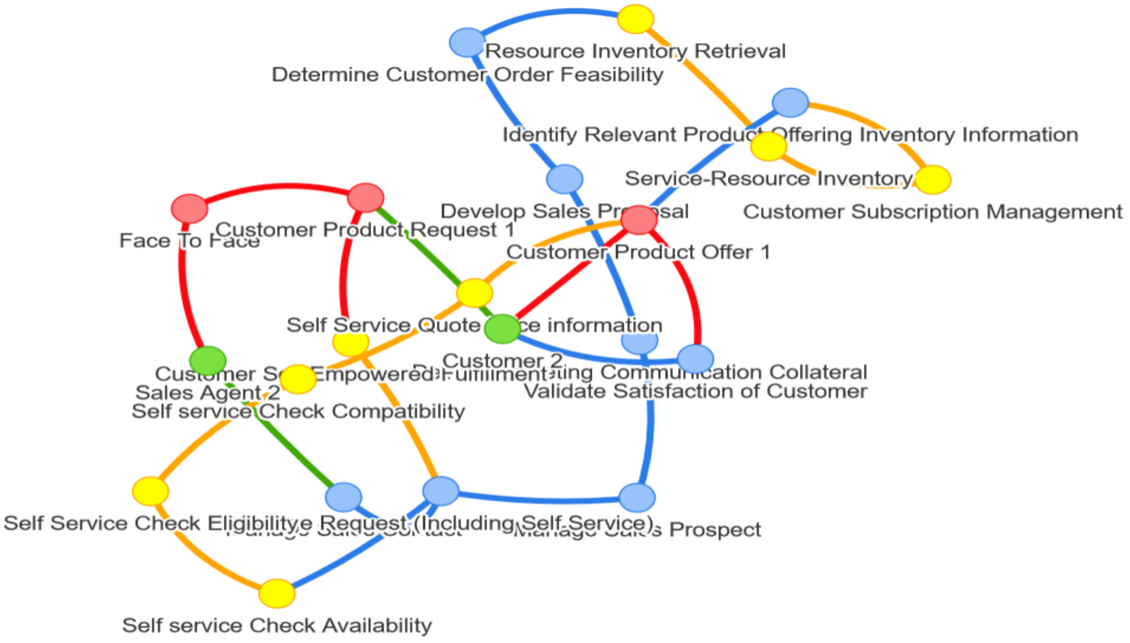
5.3.2 360-degree Scan for a node with X depth limit

The Spinning Tree Algorithm, BFS and DFS approaches are adopted in this use case. When user inputs a node, the system will traverse the graph path from the node and display all the surrounded node1s and relationships until a depth limit is reached. For example, system will return below graph result if user performs 360-Degree scan for the node "Selling" with depth limit 1.



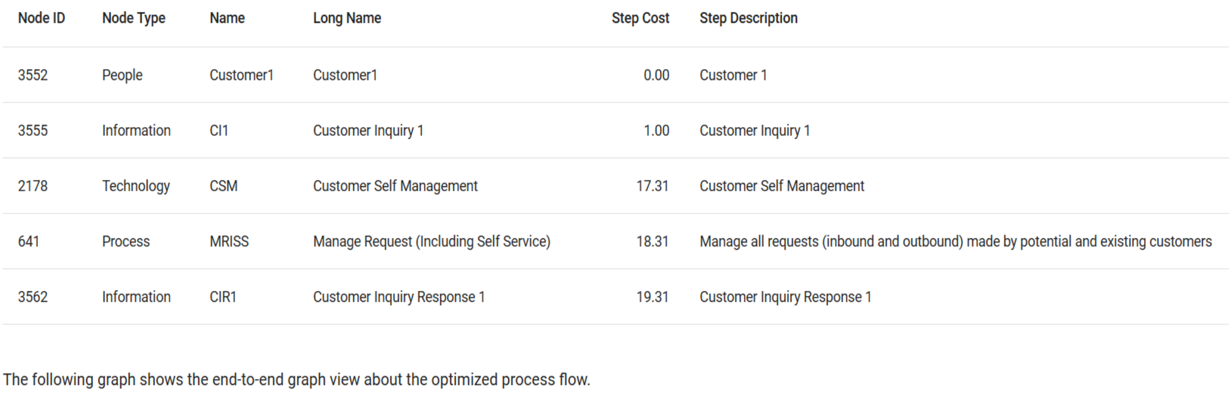
5.3.3 Relationship Scan & End-to-End Process Scan

The customized graph search query is used for this use case, scan all the nodes that are linked to a certain relationship or End-to-End business process stream according to the knowledge representation in the graph database.


Resource Inventory Retrieval
Determine Customer Order Feasibility
Identify Relevant Product Offering Inventory Information
Service-Resource Inventory
Customer Subscription Management
Develop Sales Proposal
Face To Face
Customer Product Request 1
Customer Product Offer 1
Self Service Quote
information
Customer 2
Empowered
Fulfillment
Communication Collateral
Validate Satisfaction of Customer
Sales Agent 2
Self service Check Compatibility
Self Service Check Eligibility
Request (Including Self Service)
Prospect
Self service Check Availability

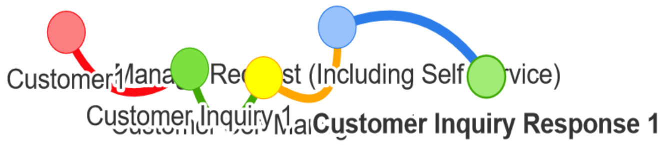
5.3.4 Recommend solution to fulfil Request-to-Answer process with lowest cost or shortest response time

The Shortest Path Algorithm (Dijkstra) is adopted in this use case. The search space is a graph model that represents an End-to-End business process stream. In the search space, system will identify the process by the relationship “Request-to-Answer”, and identify the start and end point of this process by the node property that equals to “Start” or “End” state. In the meantime, the system will calculate path cost by checking the step cost from the Cost or Time relationship properties in the graph path, and use Dijkstra algorithm to determine the shortest path with lowest cost or shortest response time. Please refer to below screenshot for the result summary and graph view of the solution.



Node ID	Node Type	Name	Long Name	Step Cost	Step Description
3552	People	Customer1	Customer1	0.00	Customer 1
3555	Information	CI1	Customer Inquiry 1	1.00	Customer Inquiry 1
2178	Technology	CSM	Customer Self Management	17.31	Customer Self Management
641	Process	MRISS	Manage Request (Including Self Service)	18.31	Manage all requests (inbound and outbound) made by potential and existing customers
3562	Information	CIR1	Customer Inquiry Response 1	19.31	Customer Inquiry Response 1

The following graph shows the end-to-end graph view about the optimized process flow.



5.4 Simulation &Distribution of Dynamic Load and Path Cost Re-calculation

As our system yet includes transactional data from existing companies, so we simulate the transactional data, in order to recommend solution in a dynamic environment. There is a developed feature for user to input the process load and resource distribution, we can implement algorithms to calculate the cost and time properties for each node in the path and observe a solution change in different loads and resource distributions.

Some defined business rules and available cost function support to calculate the cost and time that are required by each node. Please refer to below rule definition which are based on our domain knowledge and work experience.

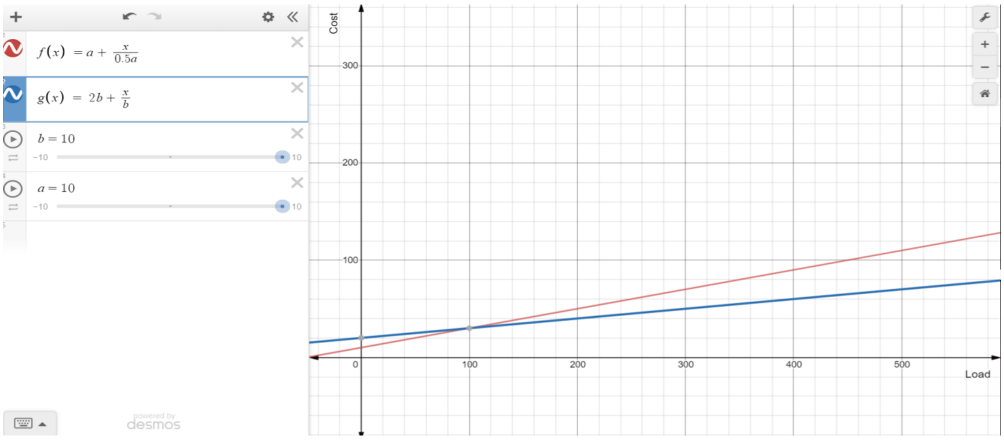
Rule1: Cost for People f(x)
IF node label is People
Then cost function f (x) = a+x/0.5*a
(a stands for value of resource property for People node)

Rule2: Cost for Technology g(x)
IF node label is Technology
Then cost function g(x) = 2b+x/b
(b stands for value of resource property for Technology node)

Rule3: Cost for Information t(x)
IF node label is Information
Then cost t (x) = 1 by default, because Information is a virtual entity

Rule4: Cost for Process z(x)
IF node label is Process
Then cost function z (x) = c+x/0.5*c
(a stands for value of resource property for Process node)

The following function graph shows the trending of the cost function. As we can see, the cost requires by People node is initially lower than Technology and Process nodes, but its increasing rate is faster than the Load. This trend indicates that if the Load is higher, the more efficient solution is to go through process by Technology rather than People.



$f(x) = a + \frac{x}{0.5a}$

$g(x) = 2b + \frac{x}{b}$

$b = 10$

$a = 10$

This is the cost function which can calculate response time for each node according to different Loads and Resource properties.

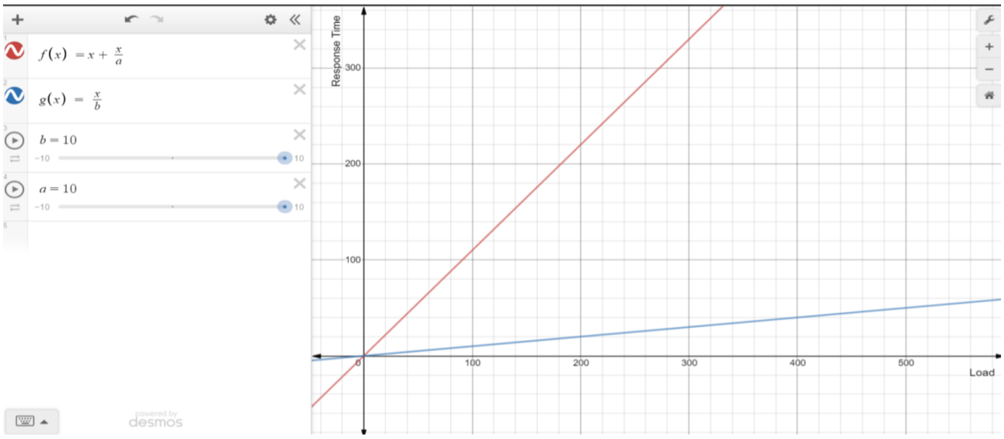
Rule5: Response Time for People f(x)
IF node label is People
Then time f (x) = x+x/a (a stands for value of resource property for People node)

Rule6: Response Time for Technology g(x)
IF node label is Technology
Then time g (x) = x/b (b stands for value of resource property for Technology node)

Rule7: Response Time for Information t(x)
IF node label is Information
Then time t (x) = 1 by default, because information is a virtual entity.

Rule8: Response Time for People z(x)
IF node label is Process
Then time z(x) = x+x/a (a stands for value of resource property for Process node)

The following function graph shows the trending of the cost function. The response time requires by People is much higher than Technology's when the load increases, this trend indicates that if the Load is higher, the more efficient solution is to go through process by Technology rather than People.



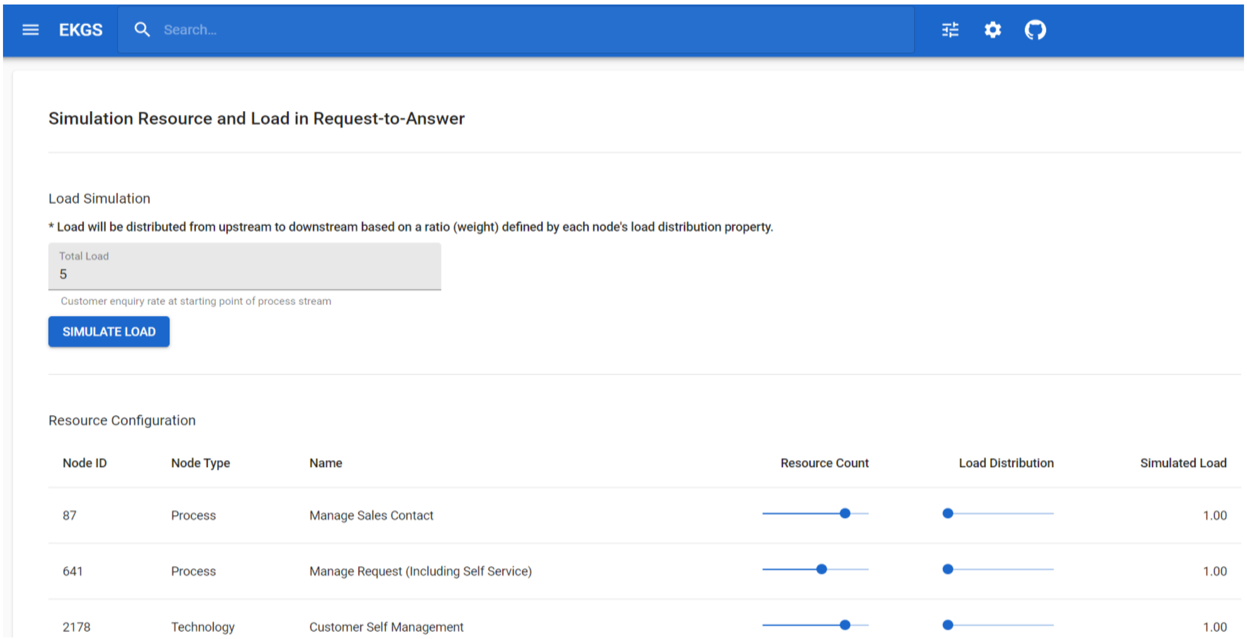
$f(x) = x + \frac{x}{a}$

$g(x) = \frac{x}{b}$

$b = 10$

$a = 10$

Furthermore, users can adjust the no. of load and resource of the whole process on page below, once the number of loads is entered in the Load Simulation Box, system will use BFS algorithm to traverse the graph and distribute the simulated Load to all nodes and relationships along the path. From the start node to the end node of the process, system can re-calculate the Cost and Time property for each node according to the above-mentioned business heuristic rules and cost functions.



Simulation Resource and Load in Request-to-Answer

Load Simulation

* Load will be distributed from upstream to downstream based on a ratio (weight) defined by each node's load distribution property.

Total Load
5

Customer enquiry rate at starting point of process stream

SIMULATE LOAD

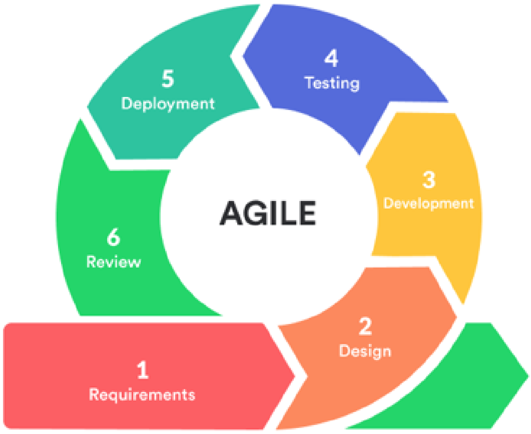
Resource Configuration

Node ID	Node Type	Name	Resource Count	Load Distribution	Simulated Load
87	Process	Manage Sales Contact			1.00
641	Process	Manage Request (Including Self Service)			1.00
2178	Technology	Customer Self Management			1.00

6.0 Project Implementation (To detail system development & testing approach.)

6.1 Agile Development & Continuous Delivery

We adopted Agile software development approach in project implementation. With such approach, application development can start early while we were still brain-storming new use-cases and baking solution designs. Application is always in ready-to-deploy state, with incremental functionality updates being gradually merged into it. As a result, testing and experiment can be carried out against the real-time application, so that defects and improvements can be identified at early stage.



To facilitate the Agile development and support continuous delivery, we have followed below practices:

- Conduct weekly team meetings via Microsoft Teams to discuss solution designs, review development progress and feedback testing results.

- Divide application into 4 services (Graph Data Service, Dialogflow Service, Back-end Service and Front-end Service), enabling each service to be developed and tested independently.
- For local services hosted in Docker, regularly publish the Docker images to Docker Hub registry to allow all team members access and evaluate the latest version of application.
- For Dialogflow service hosted in Google Cloud, regularly push graph data changes to the cloud to update training phrases and entity values.

As mentioned, the application is divided into 4 services, following chapters will discuss implementation details in each of them.

6.2 Graph Data Service

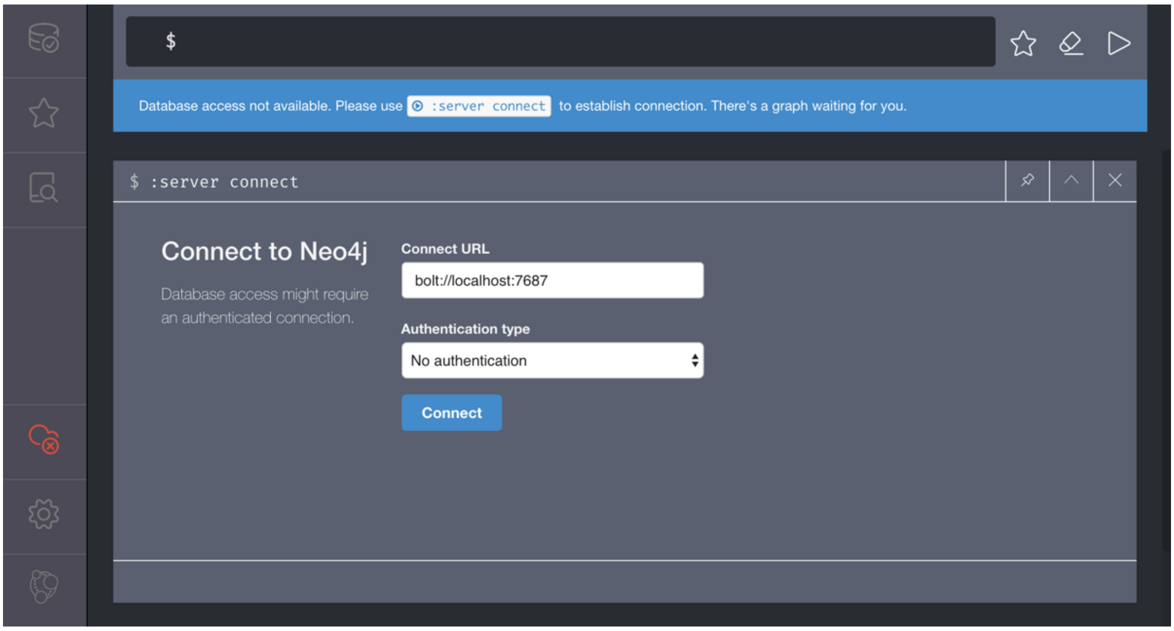
Base Docker image - neo4j:3.5.17 (URL: https://hub.docker.com/_/neo4j)
Project Docker image - irs3y/ekgs-graphdata
(URL: https://hub.docker.com/r/irs3y/ekgs-graphdata)

As the outcome of knowledge acquisition process, our application data is in CSV format and ready to be loaded into Neo4j database. In order to automate this process for each deployment, we have built a customized Neo4j Docker image which use initialization script to load all data at start-up of Docker container.

Also, to support the graph reasoning query mentioned in application design, two Neo4j plugins are pre-loaded into the customized image:

- APOC v3.5.0.9 (URL: https://neo4j.com/docs/labs/apoc/current/introduction/)
- Graph Algorithms v3.5.14.0 (URL: https://neo4j.com/docs/graph-algorithms/current/)

Graph Data Service exposes two interfaces. The 1st interface, listening at port 7687, is for Back-end Service to query and update graph data using Neo4j's Cypher query language via bolt protocol. The 2nd interface, listening at port 7474, is Neo4j's built-in web browser. As a back-door developer tool, Neo4j Browser is where we can evaluate run-time application data by manually triggering Cypher queries.



```
$
```

Database access not available. Please use :server connect to establish connection. There's a graph waiting for you.

```
$ :server connect
```

Connect to Neo4j

Database access might require an authenticated connection.

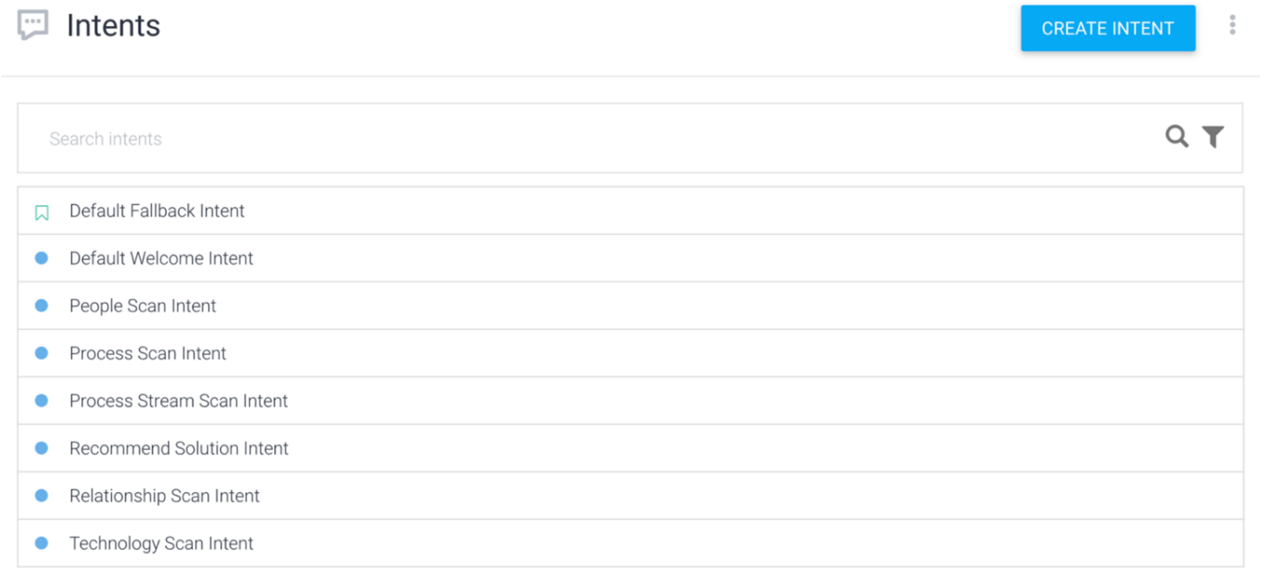
Connect URL: bolt://localhost:7687

Authentication type: No authentication

6.3 Dialogflow Service

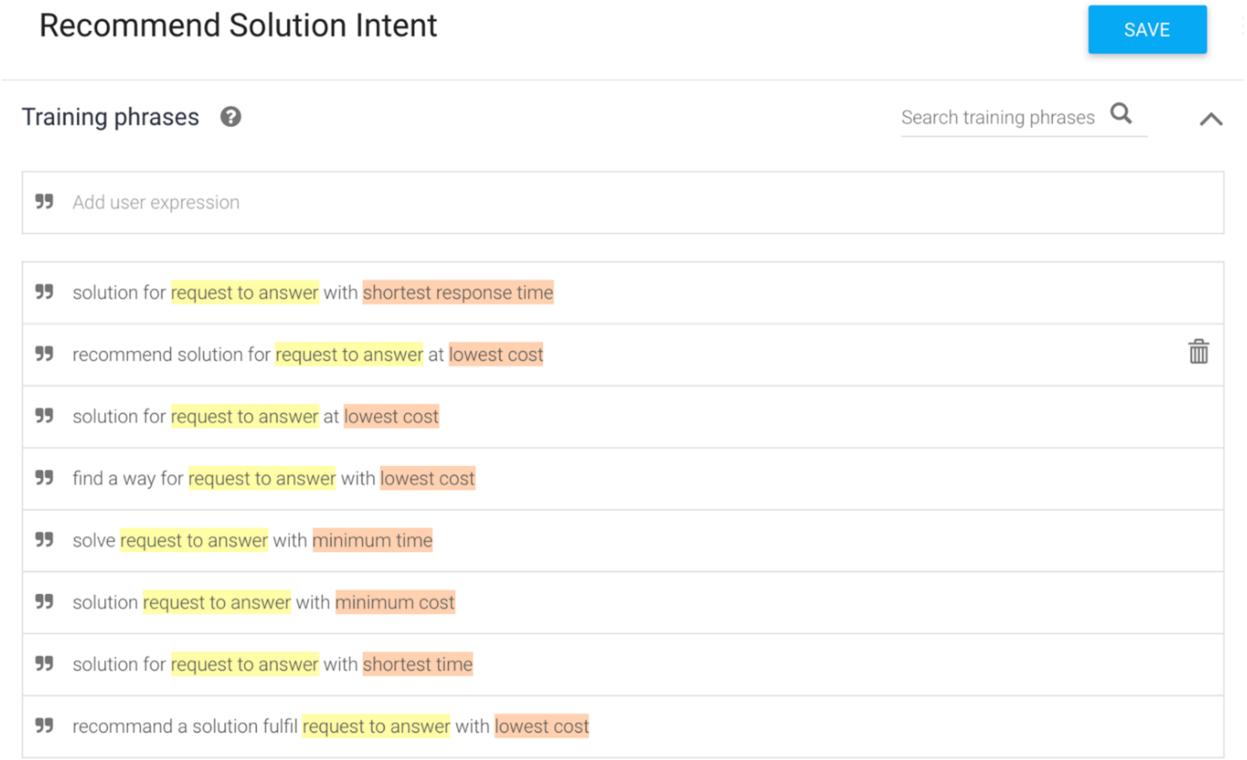
We built integration to Google Dialogflow service. The service complements search suggestion rules in the Back-end Service, and provides better user experience by enhancing the capability of processing natural language in user's search inputs. Because Dialogflow Service has to be hosted in Google Cloud, it is activated in application only when Back-end Service (Docker container) has public internet connection.

Intent definition is added to Dialogflow for each use-case we want to support.



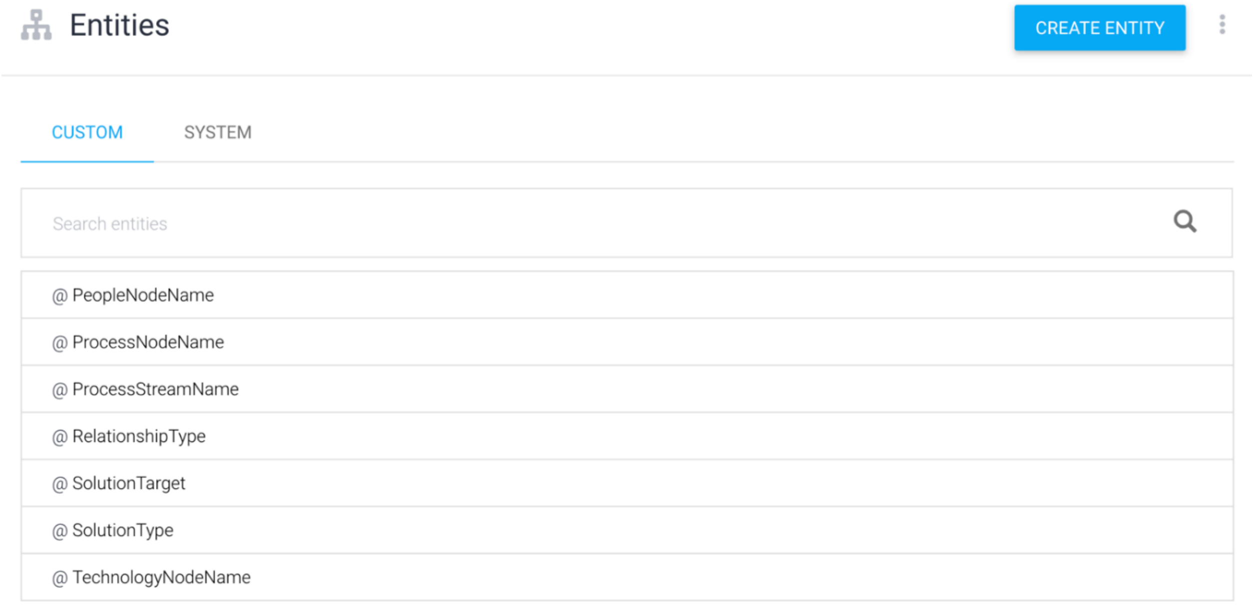
- Default Fallback Intent
- Default Welcome Intent
- People Scan Intent
- Process Scan Intent
- Process Stream Scan Intent
- Recommend Solution Intent
- Relationship Scan Intent
- Technology Scan Intent

And each intent has its own training phrases that can distinguish it from other intents. These training phrases follow the same pattern for single intent, e.g. prepend keyword 'solution' in the training phrases for Recommend Solution Intent.



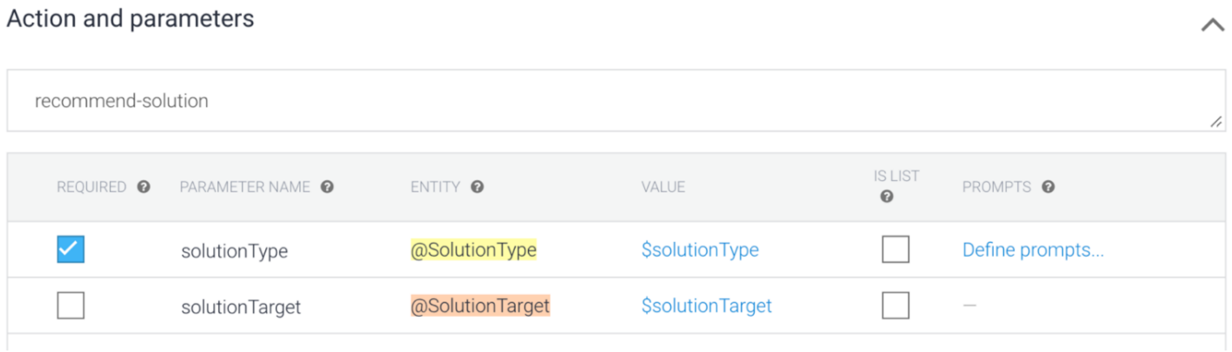
- solution for request to answer with shortest response time
- recommend solution for request to answer at lowest cost
- solution for request to answer at lowest cost
- find a way for request to answer with lowest cost
- solve request to answer with minimum time
- solution request to answer with minimum cost
- solution for request to answer with shortest time
- recommand a solution fulfil request to answer with lowest cost

To further improve the accuracy of intent detection, and support synonyms for essential search parameters, we have added custom entities for text values, e.g. node names, relation types and etc. Some of these entities, such as node names, have too many values to be manually entered into Dialogflow. We have built utility class in Back-end Service for these entities, which will query graph database and generate batch JSON files. These JSON files follow the same format as Dialogflow entity export files and can be imported into Dialogflow.



@ PeopleNodeName
@ ProcessNodeName
@ ProcessStreamName
@ RelationshipType
@ SolutionTarget
@ SolutionType
@ TechnologyNodeName

When user inputs matched an intent, action name associated with this intent and action parameters extracted from user inputs will be passed to Back-end Service for matching pre-defined rules.



recommend-solution

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	solutionType	@SolutionType	$solutionType	□	Define prompts...
□	solutionTarget	@SolutionTarget	$solutionTarget	□	–

6.4 Back-end Service

Base Docker image - openjdk:11.0.6-buster (URL: https://hub.docker.com/_/openjdk)
Project Docker image - irs3y/ekgs-backend
(URL: https://hub.docker.com/r/irs3y/ekgs-backend)

The Back-end Service is a single Java process built with Spring Boot framework. It has following architectural functions:

- Integrate with Graph Data Service (with Spring Data Neo4j libraries)
- Integrate with Dialogflow Service (with Google Cloud Java Client libraries)

- Embed Drools Rule Engine (with KIE Drools libraries)
- Provide REST APIs for Front-end Service to consume (with Spring MVC)

6.4.1 Rule Engine

Because our application only utilise rules engine feature provided in KIE, we chose to embed Drools inside Back-end Service and eliminate the implementation overhead of having full integration with KIE Workbench. Adding rules in Back-end Service is as simple as putting Drools DRL file into Java resources folder.

There are mainly three types of rules in our application:

1) Search Suggestion Rules: They are responsible for getting information from Graph Data Service and suggesting what can be searched based on current user inputs.

   e.g.

```
rule "relation scan suggestion"
activation-group "search-suggestion"
salience 10
when
  SearchInput( value != null && value.startsWith("Relationship Scan") )
  sr : SearchResults()
then
  Frameworx.searchSuggestionForRelationScan(sr, "Relationship Scan for ");
end
```

2) Search Result Rules: They are responsible for performing action (e.g. formulating Cypher query) based on intent detection result getting from Dialogflow Service.

   e.g.

```
rule "recommend solution result (dialogflow)"
activation-group "search-result"
salience 10
when
  da : DialogAction( name == "recommend-solution" )
  sr : SearchResults()
then
  Frameworx.searchResultForRecommendSolution(sr, da);
end
```

3) Cost Function Rules: As mentioned in system design, our application has load simulation function. These rules are responsible for calculating node cost based on simulated load.

   e.g.

```
rule "people node cost"
activation-group "cost"
salience 10
dialect "mvel"
when
  info : ResourceLoadCostInfo( label == 'People' )
then
  info.cost = info.resource + 2.0 * info.load / info.resource;
  info.time = info.load + info.load / info.resource;
```

end

6.4.2 REST APIs

Back-end Service provides a set of APIs for Front-end Service to perform query or update to the whole application.

1) Query Application Status
   Endpoint: HTTP GET /backend/app/status
   This API allows front-end to check back-end connectivity to Dialogflow Service.

2) Query Application Settings
   Endpoint: HTTP GET /backend/app/settings
   This API allows front-end to get system-wide settings, e.g. Graph Data Service connection information

3) Save Application Settings
   Endpoint: HTTP POST /backend/app/settings
   This API is for modifying system-wide settings.

4) Search Request
   Endpoint: HTTP POST /backend/search
   This API is for triggering search-suggestion and search-result rules when user changes search bar input.

5) Search Request (for Dialogflow)
   Endpoint: HTTP POST /backend/search/dialog
   This API complements normal Search Request API when Dialogflow Service is available. It triggers intent detection request to Dialogflow Service and generate search result based on matched intent.

6.5 Front-end Service

Base Docker image - nginx:1.15.12 (URL: https://hub.docker.com/_/nginx)
Project Docker image - irs3y/ekgs-frontend
(URL: https://hub.docker.com/r/irs3y/ekgs-frontend)

The Front-end Service is a Single-page Application (SPA) built with React and Material UI framework. The major component of UI is the search bar. All use-cases of our application are built it. When user starts to input text into the search bar, it triggers Back-end Service Search Request API, which in-turn triggers various reasoning and functional requests to Rules Engine, Dialogflow Service and Graph Data Service.

Other miscellaneous functions in front-end UI include:

- Remember historical search and append to search suggestion
- Alert Dialogflow Service disconnected status
- Change Graph Data Service connection settings
- Change Graph display settings (e.g. showing node labels)
- Simulate transaction load in process stream

For more details on user interaction with our application, please refer to the User Guide.

7.0 Project Performance & Validation

In order to prove project objectives are met, we designed following test cases, and performed in the configuration below. All test cases passing without issues.

Configuration:
SW: Win10 64bit
Browser: Google Chrome 81.0.4044.113
Docker: Docker Desktop 19.03.8, Docker-Compose 1.25.4

Test Flow	Step No.	Step Description	Expected Result	Actual Result
Docker Installation	1	Download Docker Desktop/ Toolbox and install it on laptop.	It can be installed successfully.	Pass
Trigger Web Business System	2	Open a browser, input the IP: http://localhost *Note*, if you installed Docker Toolbox, please check its IP with command "*docker-machine ip*" on a terminal.	The system page can show up without issues.	Pass
Web Settings	3	Check all buttons, such as Resource Simulation, System settings, Project site and Graph settings.	All buttons are active and working well.	Pass
Use Case tests	4	Click Search bar.	4 use cases can be auto listed.	Pass
Use Case#1: 360-degree Scan	5	Select "360-degree Scan".	Following keywords/ phrases can show up.	Pass
	6	Continue to select one keyword, such as "for People".	More keywords/ phrases (e.g. Buyer) show up.	Pass
	7	Continue to select one keyword/ phrases, such as "Buyer".	The knowledge graph for "360-degree Scan for People Buyer" can show up, brief text summary also can display on top of graph.	Pass
	8	Refer to steps 4-6, continue to test rest options, such as "for Process", "for Technology" etc.	The knowledge graph for all options can show up, brief text summary also can display on top of graph.	Pass
Use Case#2: Relationship Scan	9	Select "Relationship Scan".	Following keywords/ phrases can show up.	Pass
	10	Select one of listed phrases, such as "Relationship Scan for Sub_Process".	The knowledge graph for "Relationship Scan for Sub_Process" can show up, brief text summary also can display on top of graph.	Pass
	11	Refer to steps 8-9, continue to test rest options.	The knowledge graph for all options can show up, brief text summary also can display on top of graph.	Pass
Use Case#3: End-to-End process scan	12	Select "End-to-End Process Scan".	Following phrases "for Request_to_Answer " can show up.	Pass
	13	Select "End-to-end Process Scan for Request_to_Answer".	The corresponding knowledge graph can show up, brief text summary also can display on top of chart.	Pass
Use Case#4: Recommend a Solution	14	Select "Recommend a Solution".	Following phrases "Fulfil Request-to-Answer" can show up.	Pass
	15	Continue to select rest phrases "with Lowest Cost" or "with Shortest Response Time".	The knowledge graph with brief summary for "Recommend a Solution Fulfil Request-to-Answer with Lowest Cost" & "Recommend a Solution Fulfil Request-to-Answer with Shortest Response Time" can show up correctly.	Pass

8.0 Project Conclusions (Findings & Recommendation)

In the beginning of this project, we had a long-term planning to build an intelligent business system, and add more feathers on the system when we gradually learn technical AI knowledge and skillsets in the future. In this semester, it is quite significant to set up the solid basis and some necessary “smart” functions. In order to put all studied theories of three courses into practice well, our team members referred the process of CRISP-DM methodology to deploy knowledge discovery, knowledge extraction, knowledge reasoning and knowledge model construction into the overall system design. Currently, the Enterprise Knowledge Graph System is developed based on a mature framework “Framworx”, its database involves 3000+ nodes and 2500+ relations, and different algorithms had been implemented successfully for each function design.

While working on the project, our team with 3 members always had very positive and efficient discussions. Although we all have to study at our own home without face-to-face communication this semester, we still learnt a lot from each other as we were keen on sharing respective professions and skillsets during weekly meetings for this project. It is a good opportunity for us not only to apply knowledge into practical project, but also a wonderful time for every team member to experiment and learn new things.

Finally, the following further enhancement will be considered to implement from next semester onwards.

- Extend more widely useful end-to-end business processes for both internal and external consumers, such as “Order-to-Payment”, “Request-to-Change” & “Termination-to-Confirmation” processes etc.
- Develop more use cases including job scheduling & field engineer route planning etc.
- Further strengthen the practical business value of the system. For example, with application of a large number of data shared by actual organizations, to customize reasonable and specific knowledge models for different kinds of requests.

Appendix A: Project Proposal

Please refer to file "EKGS_Project_Proposal_Group2.pdf".

Appendix B: Mapped System Functionalities against knowledge of Courses

Knowledge of Courses	System Functionality
Machine Reasoning ✓ Knowledge Acquisition Documented Source Undocumented Source ✓ Knowledge Discovery CRISP-DM methodology	**Knowledge Discovery** Extract enterprise operation knowledge from structured data (CSV format) and un-structured data (guideline for best-practice) from TMForum - Framworx standards, as well as our domain knowledges.
Machine Reasoning ✓ Knowledge Models (Acquire -> Represent) Decision Tree ✓ Knowledge Representation Knowledge Graph	**Knowledge Representation** Establish graph data structure based on linked nodes with labels and properties; Represent enterprise entities and process flow with path cost attributes inclusion, such as time and effort cost.
Reasoning Systems ✓ Uninformed Search Techniques Spinning Tree Algorithm (DFS & BFS) Shortest Path Algorithm (Dijkstra)	**Knowledge Inference** Adopt AI search/reasoning algorithms (Dijkstra) to generate optimal recommendation of the process/solution according to configurable node properties and path cost attributes.
Cognitive System ✓ Rule engine ✓ Decision Tree	**Cognitive System** Capture user text input and translate it into searching or reasoning rules of the system through the rule-based decision tree method.

Appendix C: Installation and User Guide

Please refer to file "EKGS_User_Guide_Group2.pdf" in details.

Appendix D: List of Abbreviations

Abbr.	Full Name
AI	Artificial Intelligence
BFS	Breadth-First-Search
CRISP-DM	Cross Industry Standard Process for Data Mining
CSPs	Communication Service Provider
DB	Database
DFS	Depth-First-Search
ETL	Extract-Transform-Load
eTOM	Enhanced Telecom Operation Map
PPT	People, Process & Technology
RDMS	Relational Database Management System
SME	Small and medium-sized enterprises
TM Forum	Renamed as "Business Process Framework" since 2013

Appendix E: References

1. Amit Singhal. 2012. Introducing the Knowledge Graph: things, not strings. [https://www.blog.google/products/search/introducing-knowledge-graph-things-not/]. Accessed 14 April 2020.
2. Tencent Knowledge Graph. [https://cloud.tencent.com/product/tkg/details]. Accessed 1 May 2020.
3. Enterprise Profile. [https://data.aliyun.com/product/eprofile]. Access 1 May 2020.
4. TM Forum FRAMWORX Model. [https://www.tmforum.org/frameworx-homepage/]. Accessed 7 May 2020.
5. Neo4j Graph Developer Resource. [https://neo4j.com/docs/]. Accessed 7 May 2020.