GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)
PRACTICE MODULE: Project Proposal

Date of proposal:

28 February 2020

Project Title:

ISS Project – Automated Triaging System, for intelligent helpdesk ticket triaging system use case

Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)*

Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE (NUS)
Contact: Mr. GU ZHAN / Lecturer & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

Background/Aims/Objectives:

Problem:

Customer service across all industries, especially IT operations management, struggle volume of incoming tickets at one point or other since the available resources to resolve them are fixed. Traditionally, the process is reactive. User calls or emails in a request or opens a ticket. Helpdesk staff or engineer sees it, gathers any additional information from the user, prioritizes it, routes it, and it is eventually addressed. There are a lot of administrative steps in that process, causing layers of inefficiencies.

Market Research:

Artificial Intelligence (AI) is increasingly making its presence known in the service industry. According to Gartner[1], half of all medium to large enterprises will utilize AI-powered chatbots in the workplace by 2020. One of the biggest benefits of AI for the help desk and overall IT support function is that it is believed to be able to remove the manual overhead associated with high-volume, low-value service desk activities.

Early AI capabilities are embedded in some existing IT service management (ISTM) product suited. ServiceNow, Micro Focus, BMC, Symphony Summit, Ivanti, IBM, ServiceAide, and Freshworks are among the vendors who have integrated AI into their products suites. However it was observed that these product suites are costly and require high capital investments to acquire the base platform.
We see a market gap in addressing the demands of small-medium enterprises and startups, where AI capabilities could be made available and affordable to integrate to their existing (usually open-source) ticketing systems.

[1] https://www.gartner.com/smarterwithgartner/chatbots-will-appeal-to-modern-workers/

Proposed Solution:

We propose to develop an intelligent system which can analyse a ticket and assign a priority according to the urgency of the issue. The API-centric product will ease integration with the customer's existing ticketing system. Customers need not invest in alternative ticket systems.

The Automated Triaging System will be developed and marketed as a AI(or software)-as-a-service.

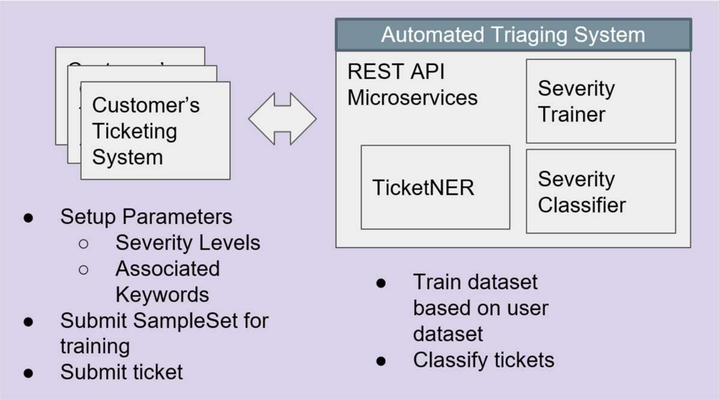
Proposed Success Measurements:

The returns on investment (ROI) on AI implementation has to be obvious to encourage early adoption and low-cost mass proliferation.

The turn-around time from the time the ticket is received to the time the customer service staff identifies the severity could be measured. The control factor is that both manual and AI-enabled processes use the same business process and rules. The cost savings could be represented by the amount of time saved per staff by the service desk.

System Details (Preliminary design)**:**

Features:



- Setup Parameters
  - Severity Levels
  - Associated Keywords
- Submit SampleSet for training
- Submit ticket

- Train dataset based on user dataset
- Classify tickets

An Intelligent ticket analysis and severity assignment system is expected to be built which will be deployed as a microservice using Django Web Framework and supporting python libraries.

Process:

1. The customer will setup the various parameters of the system such as the Severity Guidelines (Priority 1, 2, 3; Severity 1, 2, 3 & colour coding such as RED, AMBER, GREEN)
2. The customer will submit a set of previous tickets along with their initial priority and final priority will be provided to train the system

3. The system will learn through the text provided in the ticket to identify the word map which can be used in the future to associate severity with the key words.
4. Once training is completed, the system will accept one ticket data at a time and return a severity/priority value (P1 S3 etc).
5. Integration with the customer's ticket system can be implemented in future to automatically triage the incoming tickets and returns priority.

Requirements Overview:

- Research ability
- Programming ability
- System integration ability

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:

- GPU, Windows PC or a MAC, a cloud server for running models

Software proposed for consideration:

- Reasoning systems, e.g. KIE jBPM, Drools, AppFormer, OptaPlanner, Fuzzy logic, Optimization, etc
- Pertained machine learning models, e.g. Vision, Speech, NLP
- Machine learning use cases, e.g. Orange3, R
- Deep learning tools, e.g. Neural Network Console Sony, Python Keras
- Chat-bots, e.g. ChatterBot, DBpedia Chat-bot
- Cognitive systems, e.g. MyCroft
- Robotic Process Automation, .e.g TagUI
- Cloud computing/server, e.g. Amazon, Google, IBM, Azure, etc.
- Application container, e.g. Docker

Number of Learner Interns required: (Please specify their tasks if possible)

A team of two project members

Methods and Standards:

Procedures	**Objective**	**Key Activities**
Requirement Gathering and Analysis	The team should meet with ISS to scope the details of project and ensure the achievement of business objectives.	1. Gather & Analyze Requirements 2. Define internal and External Design 3. Prioritize & Consolidate Requirements 4. Establish Functional Baseline
Technical Construction	· To develop the source code in accordance to the design. · To perform unit testing to ensure the quality before the components are integrated as a whole project	1. Setup Development Environment 2. Understand the System Context, Design 3. Perform Coding 4. Conduct Unit Testing

Integration Testing and acceptance testing	To ensure interface compatibility and confirm that the integrated system hardware and system software meets requirements and is ready for acceptance testing.	1. Prepare System Test Specifications 2. Prepare for Test Execution 3. Conduct System Integration Testing 4. Evaluate Testing 5. Establish Product Baseline
Acceptance Testing	To obtain ISS user acceptance that the system meets the requirements.	1. Plan for Acceptance Testing 2. Conduct Training for Acceptance Testing 3. Prepare for Acceptance Test Execution 4. ISS Evaluate Testing 5. Obtain Customer Acceptance Sign-off
Delivery	To deploy the system into production (ISS standalone server) environment.	1. Software must be packed by following ISS's standard 2. Deployment guideline must be provided in ISS production (ISS standalone server) format 3. Production (ISS standalone server) support and troubleshooting process must be defined.

Team Formation & Registration

Team Name: **TR(AI)GE**
Project Title (repeated): **Automated Triaging System**
System Name (if decided): trAIge
Team Member 1 Name: **Kumaravelu Varadharajan**
Team Member 1 Matriculation Number:
Team Member 1 Contact (Mobile/Email): 93873665/k.varadharajan@gmail.com
Team Member 2 Name: **Tng Sian Soo (Shawn)**
Team Member 2 Matriculation Number:
Team Member 2 Contact (Mobile/Email): **97926928 / shawntng@gmail.com**

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		