IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 10, October 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

Webservices Learning Bot: A Responsive, Actionable AIML BOT Framework

Akash Singh[1], Srikari Rallabandi[2], Narayana Rallabandi[3]

Associate Consultant, InvenioLSI, Hyderabad, India[1]

UG Student, Vidya Jyothi Institute of Technology, Hyderabad, India [2]

Senior Principal Architect, InvenioLSI, Hyderabad, India [3]

ABSTRACT:Conversational systems and chat-bots are evolving over a period of time and are at the verge of entering into decision making systems. The raise of Micro services that are API based applications based on standards like RAML, JSON are request-response based. Messaging Applications like WhatsApp, Telegram are offering both push notifications as well as API invocation capabilities as Actions. In this paper we discussed the possibility of a ML based ALICE BOT (As an extension to Program AB) which can learn over the RAML or WSDL or Open API or rsyslog that can be adaptable at commercial level which can invoke "pluggable" actions either for messaging or escalate based workflows that can be responsive and can be helpful in decision making. The solution we discussed is a framework that is extendable and adoptable with learning over other standards apart from the ones mentioned as well as the Actions Framework which can plug-in any custom action as per the need of the application.

ACKNOWLEDGEMENT: We would like to extend our sincere thanks to Razia Ali for assisting and helping us throughout the process of writing this manuscript.

KEYWORDS: Conversational AI, Chatbot, Natural Language Processing

I. INTRODUCTION

Our world is rapidly moving towards the era of Artificial Intelligence and considering the business systems which are based on automation. We are adapting technology which can interact like humans, extract knowledge, and make decisions with the assistance of knowledge-based systems to automate all possible activities. Digital assistants or AI based chat-bots are the most widely available solution in the field of automation monitoring. Governments also started using chat-bots as part of e-governance and standard query answering. WHO's Chat-bot as part of COVID-19 management is an example of that. Modern conversational systems started inundating our lives with devices like Amazon Echo and Google Home, and social robots like Pepper and Furhat. As more and more industries are adopting the conversational systems for reducing human dependence and automation of the regular tasks[1]

Since we already have the AI based and Rule based bots available in the market which are quite efficient in the domain of management, hospitality and software applications, our idea is to develop an Intellectual generic bot that can not only learn text and documents but also interacts with REST, SOAP and other markup language-based web and micro services. The motive behind this is to command your Infrastructure, Platform and Software as Service that are exposing web services and to make them work for you. The key advantages are time and location agnostic interactions, and possibility of proactive actions, which give the business a better Return on Investment (ROI) with monitoring and management of services at their fingertips while serving the global customers.

The prominent implementation of the AI system is a "chat-bot" which can simulate the human brain, process natural-language text and work in the field of neural networking. A chat-bot is a software application that is used to interact with computers and prepare responses like an intelligent entity to provide the essence of a live human agent. The interaction can be done through text or voice command. Since chat-bot programs are based on NLP, it interprets human language which is being fed. Some of the productive functions of chat-bots are to perform calculations, play chess and as a digital learning tool. The challenge in AI enabled digital assistants is to develop a bot which can simulate intelligence. In this paper, we will present a practical solution to develop a responsive, actionable dynamic chat-bot

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

which can learn WSDL/RAML based Web Services. The proposed bot learns on the RAML and WSDL documents generates AIML templates based on the Verb (operation) and corresponding Noun (Request - Response) pattern matching techniques. The templates used during the interaction with the bot for dynamically interpreting the "service" to be called and to provide the optimum results in the field computation using the metadata of the service generated as the response.

The proposed bot framework is a step forward in comparison with the existing bots with respect to the Actions that it can perform on the responsive ecosystem that they are representing. Since the framework offers pluggable Services (with service interface definitions) and Actions like Mail/Notification and other platforms like WhatsApp and Telegram etc. The BOTs built out of the framework with an automated system escalation matrix can be used across DevOps dashboards, as mobile applications or can be integrated into any other ecosystem which requires 24/7 availability and quick response w.r.t. service availability and the custom remedial actions that can be taken without manual intervention.

II. RELATED WORK

There are numerous research works conducted in former times which evolves around conversational AI based automation, monitoring and chat-bots[2]. This section is focused on elaborating the various theories and related work that are relevant to the chat-bot. There are two main categories to classify chat-bots: One is based on a set of rules and other uses artificial intelligence. A renowned example of chat-bot is the ALICE Bot (Artificial Linguistic Internet Computer Entity), which utilizes AIML pattern matching techniques[3]. In this, a computer program (can be either Java or Python) is used to convert the machine-readable text to the AIML format and using the pattern matching techniques, the set of chat-patterns is being utilized by ALICE to assist as tools in the various domains. Mohinish Daswani et. al. proposed a conversational AI chat-bot using algorithms based on NLP and deep learning[4]. For training the model, semantic similarity and sequence-to-sequence learning techniques are being used. This ALICE integrated College Bot can maintain context in a user session and retrieve the appropriate answers to subsequent, semantically related questions, but to achieve this a rich data set is required to train the model. Patchara Vanichvasin[5] found out that chat-bot can be used as a digital learning tool to increase efficiency in the research domain. The chat-bot was developed from the literature review of chat-bot technology and content of research knowledge with strong focus on the content, organization of the content and the application of chat-bot.

There are some domain specific chat-bots as well like BANK CHATBOT that offers 24/7 services to its customers to enhance productivity and solve bank related queries[6] and customer service escalations[7]. Its fast response reduces the time cost for both bank and customers. This involves the process of applying NLP on the user input, using ML classification algorithm to identify the class it belongs to, and then finally applying cosine similarity between the input query and the questions of that same identified class. Martin Adam et. al. performed a survey on AI-based chat-bots used in customer service and demonstrated its effects on user compliance[8]. The study is based on an online experiment that depicts the utilization of AI enabled software agents and its efficiency in terms of time and cost. The underlying data structure that most of the chat-bot is built upon is AIML. Artificial Intelligence Markup Language (AIML) is an XML dialect for creating Natural Language software agents like chat-bots. Yungang Wei et. al. performs a comparative analysis on AIML and its vital application in the field of chat-bot technology[9].

Ashikhmin, Nikita et. al. proposed another method to achieve computation in the field of web service is by using RAML[10]. In this, they have mocked the web services using RAML specification and validate the request based on the predefined format of parameters. The response body is created by parsing the JSON schema file and then generating the response specified in the RAML file.

Pablo Ca˜nas et. al. proposed a method for versatile conversations with data-driven dialog management based on machine learning techniques[11]. Task-oriented dialog systems (a.k.a slot-filling systems – they fill a data structure with a set of slots required to complete the transactions) engage an interaction with the users by means of asking them a series of questions to complete a specific task instead of engaging them in general conversational interaction in comparison to chat-bots. Users can interact with flexibility in order to formulate their queries to the system and provide additional information to the one strictly required by the system prompts. They integrated the proposal into the Dialog Flow platform, easily assembling the set of components in the interface, and deploying a ready-to-use application that can be integrated into different environments and devices.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

Gómez-Albarrán, Mercedes et. al. proposed an approach to the production of web-based interactive fiction, which is grounded in the requirements posed by an expert focus group, and which integrates a domain-specific language (DSL) for interactive fiction (HEXIFE) and an authoring tool for this DSL (IFDBMaker)[12]. HEXIFE is an extension of HMTL5 that includes markup specifically devoted to different aspects of interactive fiction, such as hyperlinking, choice points, personalization, stretch text, annotations, conditional content, and gamification.

Eleni Adamopoulou et. al. in their paper "An Overview of Chatbot Technology" proposed Classification based on the input processing and response generation method takes into account the method of processing inputs and generating responses[13]. There are three models used to produce the appropriate responses: rule-based model, retrieval based model, and generative model. The retrieval-based model discussed, offers more flexibility as it queries and analyzes available resources using APIs. The proposed bot falls under retrieval based model where it retrieves responses from the API based systems. On similar lines M´arcio Alencar et. al. in their paper "Developing a 3D Conversation Agent Talking About Online Courses"[2] discussed about using chat-bot that can interact through a 3D avatar, which can move and talk to the user, with a female Brazilian voice.

Ho Thao Hien et. al. discussed smart learning environments and presented the FIT-EBot, a chat-bot, which automatically gives a reply to a question of students about the services provided by the education system on behalf of the academic staff[14].

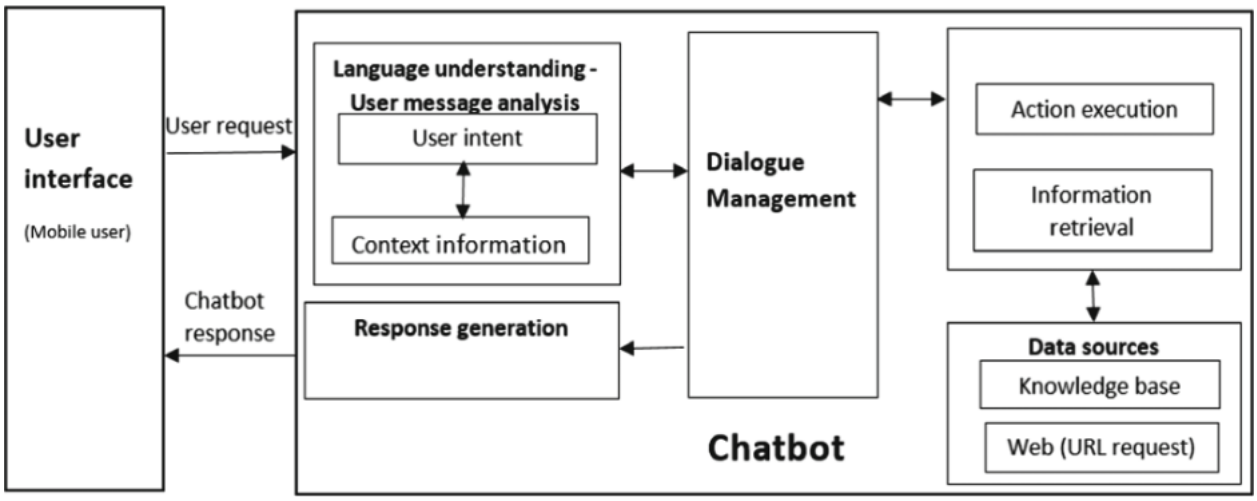

Fig: General Chatbot Architecture

All the above-mentioned techniques are either restricted to NLP or AI or specified for any domain which is not solving the purpose of generalization. On contrary, our proposed idea is the combination of all three, i.e., it uses the NLP and AIML on top of any kind of Web Service definitions offered by either WSDL or RAML or any other parse-able service definition to develop a responsive, actionable dynamic generic bot which can communicate with the web services, supports personalisation, store credentials, maintains sessions and provides user-based responses. Furthermore, it allows serverless deployment which offers scalability and flexibility at a low cost.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

III. ARCHITECTURE AND METHODOLOGY

Understanding what the chat-bot will offer and what category falls into helps developers pick the algorithms or platforms and tools to build it. The requirements for designing a chat-bot include accurate knowledge representation, an answer generation strategy, and a set of predefined neutral answers to reply when user utterance is not understood. After the chat-bot receives the user request, the Language Understanding Component parses it to infer the user's intention and the associated information. Once a chat-bot reaches the best interpretation it can, it must determine how to proceed. It can act upon the new information directly, remember whatever it has understood and wait to see what happens next, require more context information or ask for clarification.

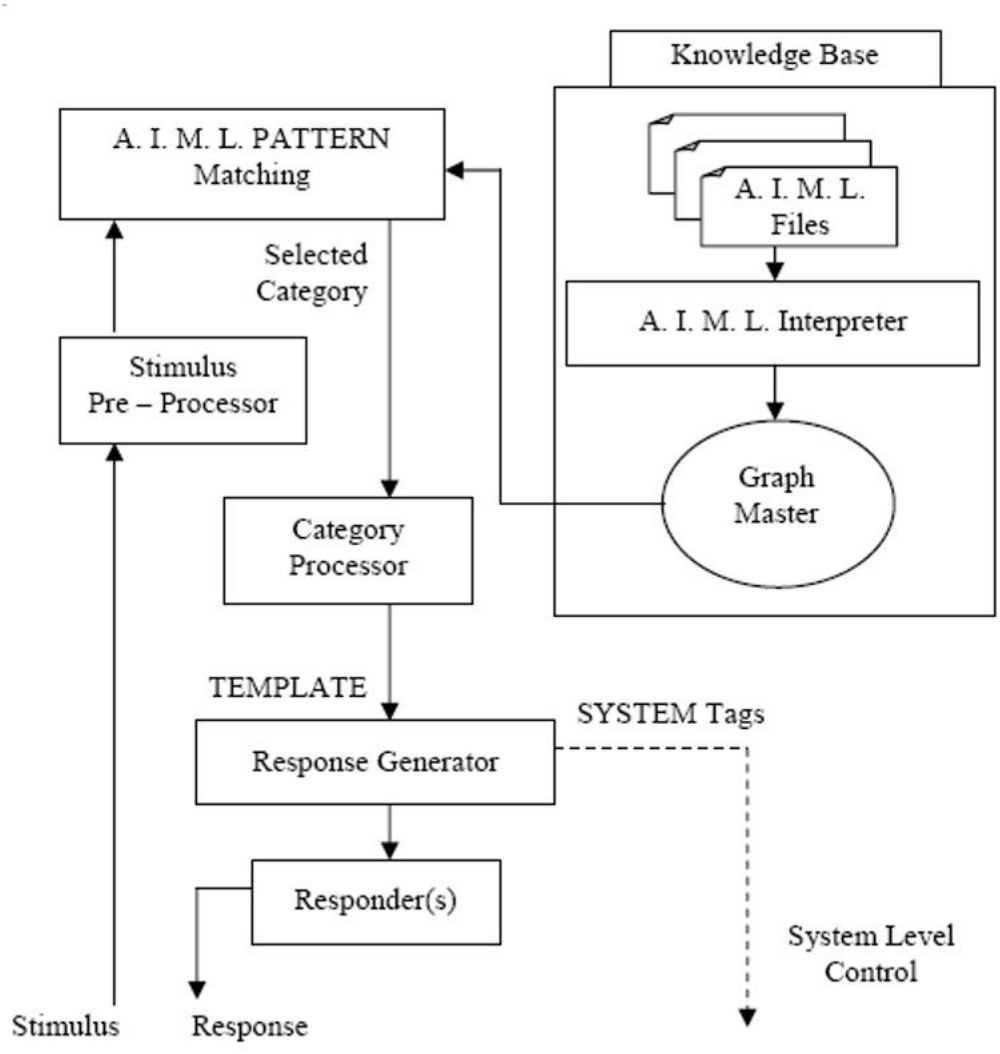

Figure 2: Structure Of AIML

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

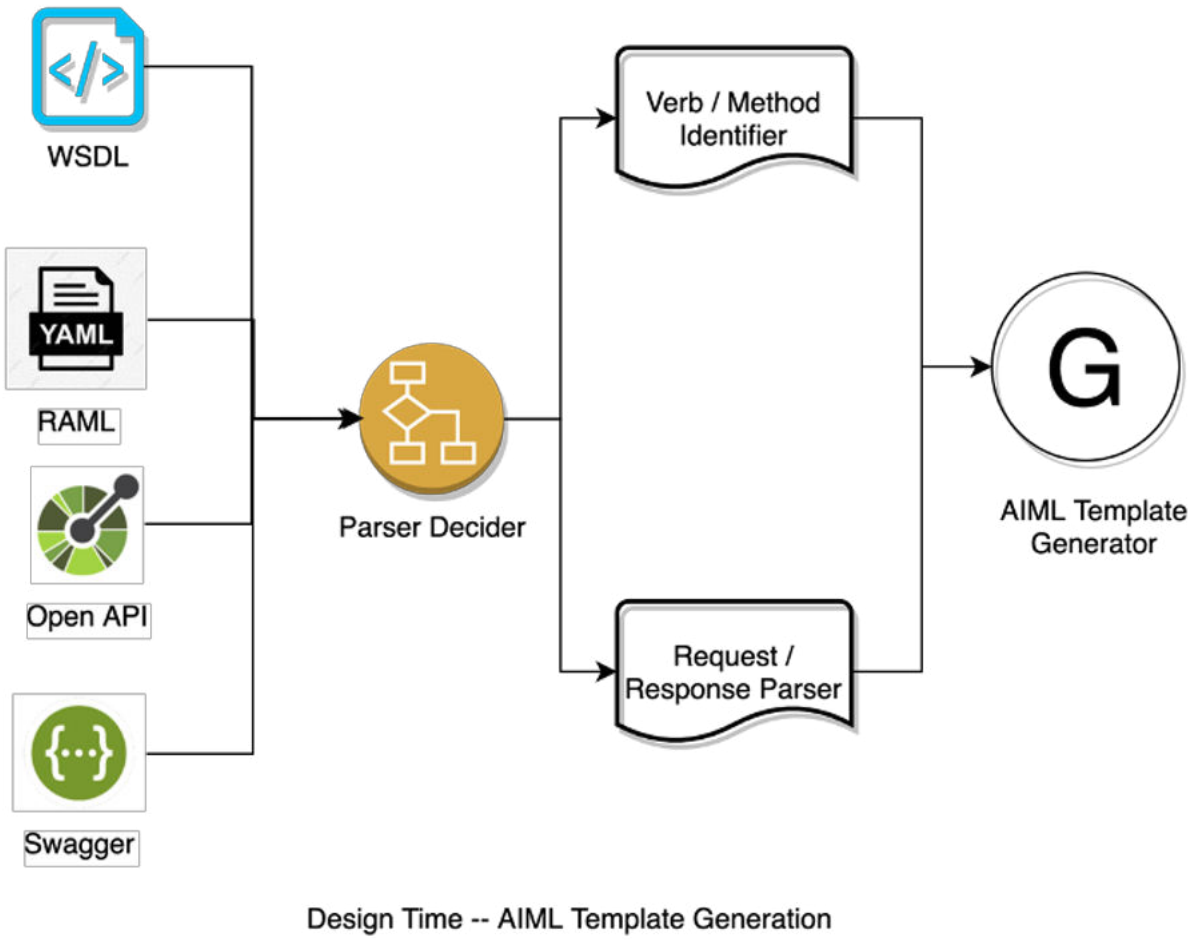

Figure 3: Template Generation component Diagram

1. **Structure of AIML:** Figure 2, shows the structure of the base conversational AIML files that are used by the Program AB. The basic conversational AIML files and rules from Program AB, to give the chat-bot ability to do out of the box conversation with the user.

2. **Structure of RAML**: A RAML crawler is used for generation of patterns, and it is built on top of Mule-RAML API. The generated pattern is used to generate a base AIML template.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

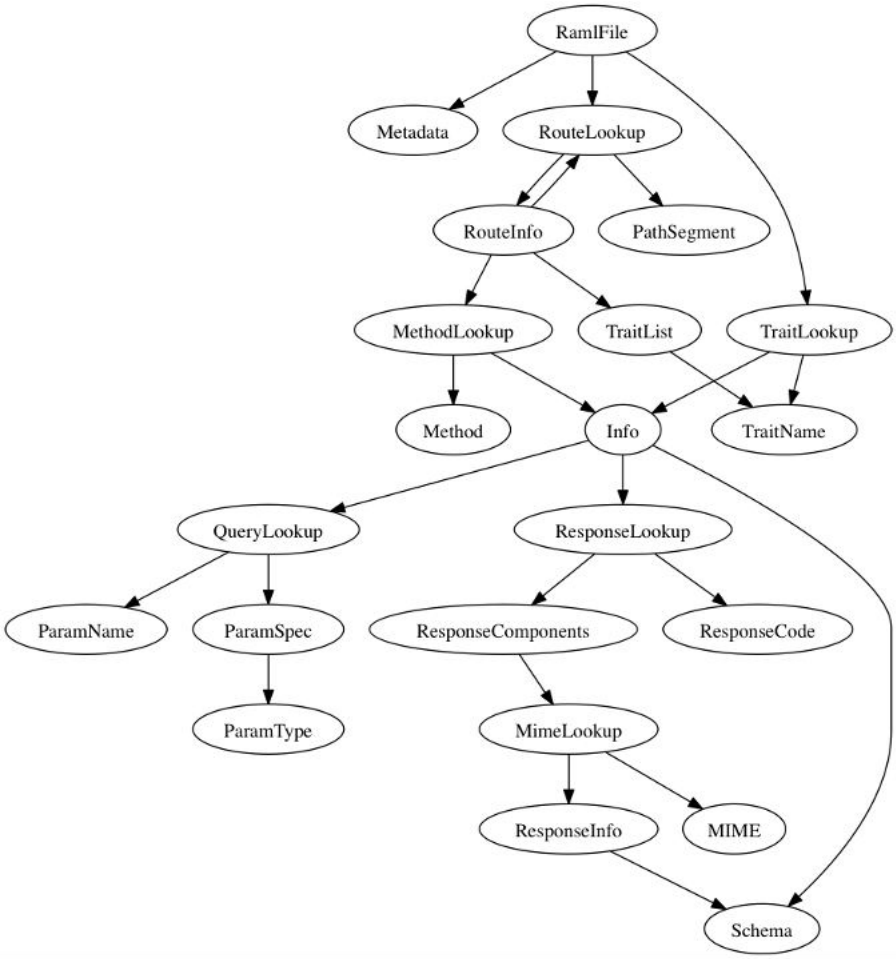

Figure 4: Structure of RAML File

3. **AIML Template Generation**: RAML API specification Crawler shown in Figure 5, has the ability to parse the specification files written in RAML, OpenAPI and Swagger, and is then provided with the specification file. The crawler then parses through the file, and generates what we call Mother AIML.

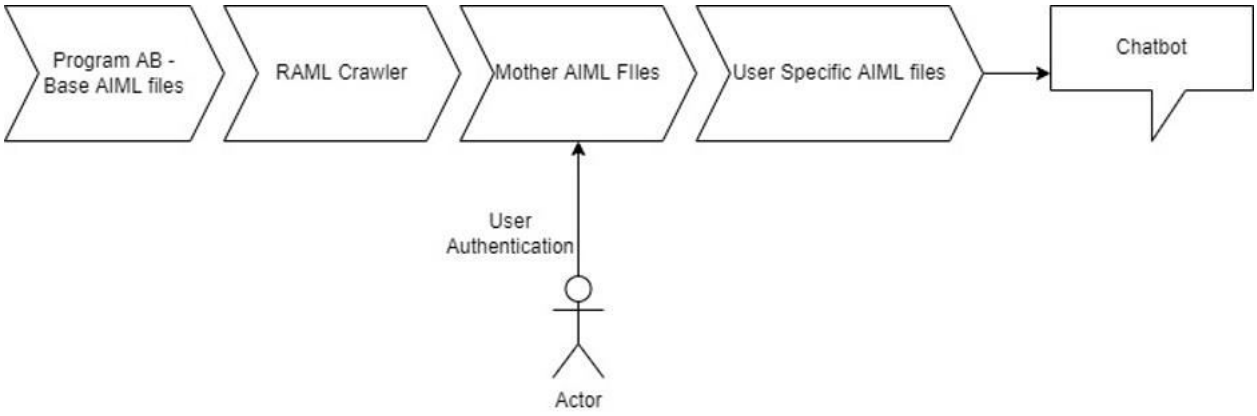

Figure 5: Custom AIML structure generation process for the RAML

4. **Algorithm for generation of the RAML Crawler**: The figure 6, shows the tree that is being created by the RAML parser after going through the API specification for the MuleSoft CloudHub API.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

The root of the tree is the base URL of the endpoints you want to parse. Each branch represents the path provided in the API specification. the leaf node represents the user intent which they want to query. These intent can be any of the HTTP methods, depending on the API specification and what that particular endpoint takes as input. Depending upon user intent, respective AIML is called by the chat-bot to provide the relevant output to the end user.

This AIML is common for all the users and contains basic information on API endpoints. All the user-specific AIML files are also created at this step, but the User Specific details are left as placeholders. Once the user authenticates their details, the user specific details are then put in place of these placeholders. The AIML files are now ready for User queries.

In this paper we are discussing a part of implementation of the proposed framework that we implemented over REST API as proof of concept (POC).

In this POC, dynamic learning is used to train the model and on top of if RAML is functioning. RESTful API Modeling Language (RAML) is a YAML-based language for describing RESTful APIs. It provides all the information necessary to describe RESTful APIs and can also describe APIs that do not fulfill all constraints of REST.

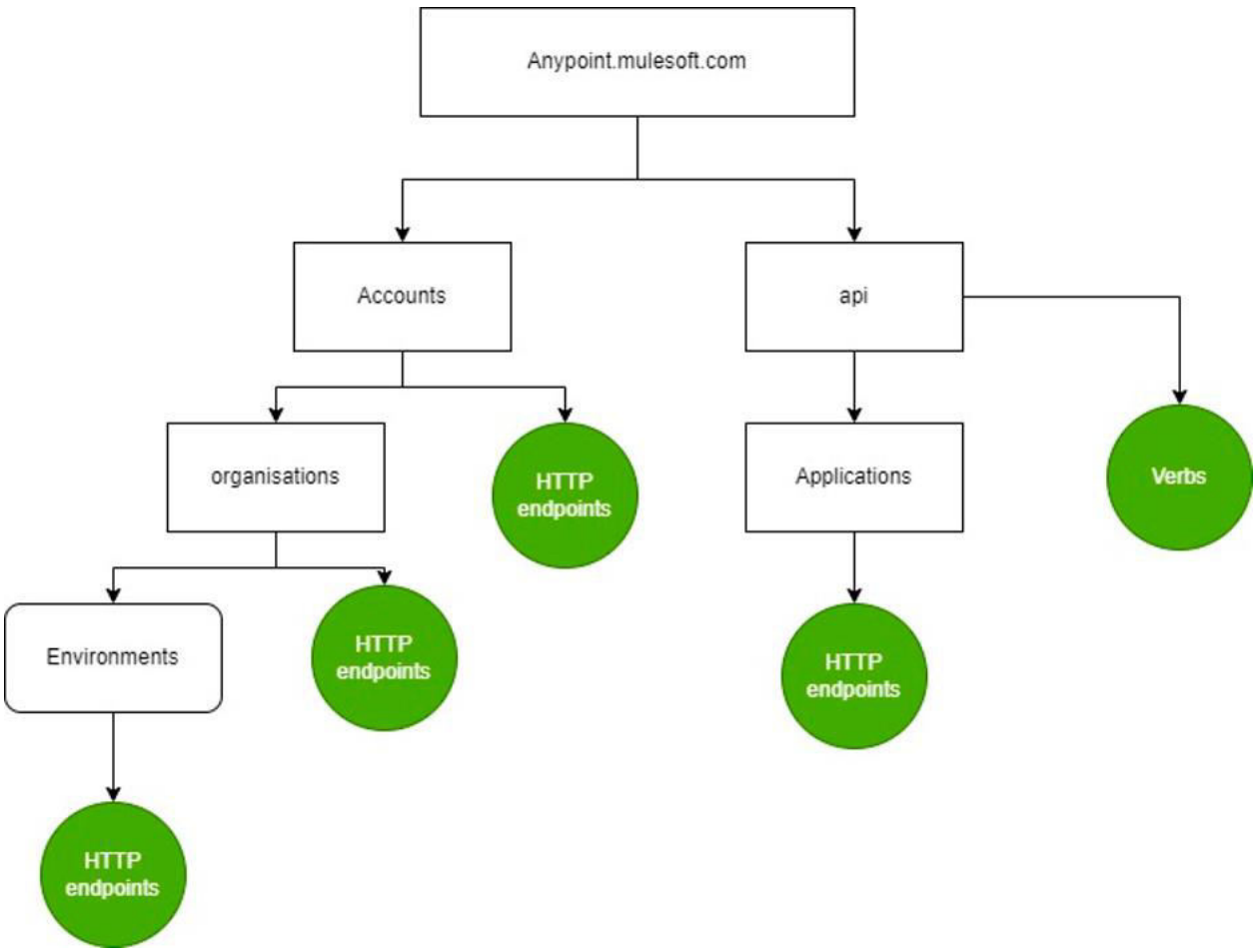

Figure 6:Hash-Map Generator for the Crawler

The User interacts with the RAML bot by using voice command or text. Once the request is received, the Chat-bot Request Handler module processes the request with the help of Natural Language Processing. In the very first step of pre-processing, the request is converted into text format and all the helping words like articles and prepositions are removed and keywords are stored. These keywords are being fed to Program AB which is built on AIML (A.L.I.C.E bot)[3]. It is internally calling the CloudHub APIs to establish the session. This module is also responsible for preparing

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

the API response in the JSON format and parsing it into the hash map to break the hierarchy. All the keys are brought to the root level to optimize the search, this is achieved by level order traversal.

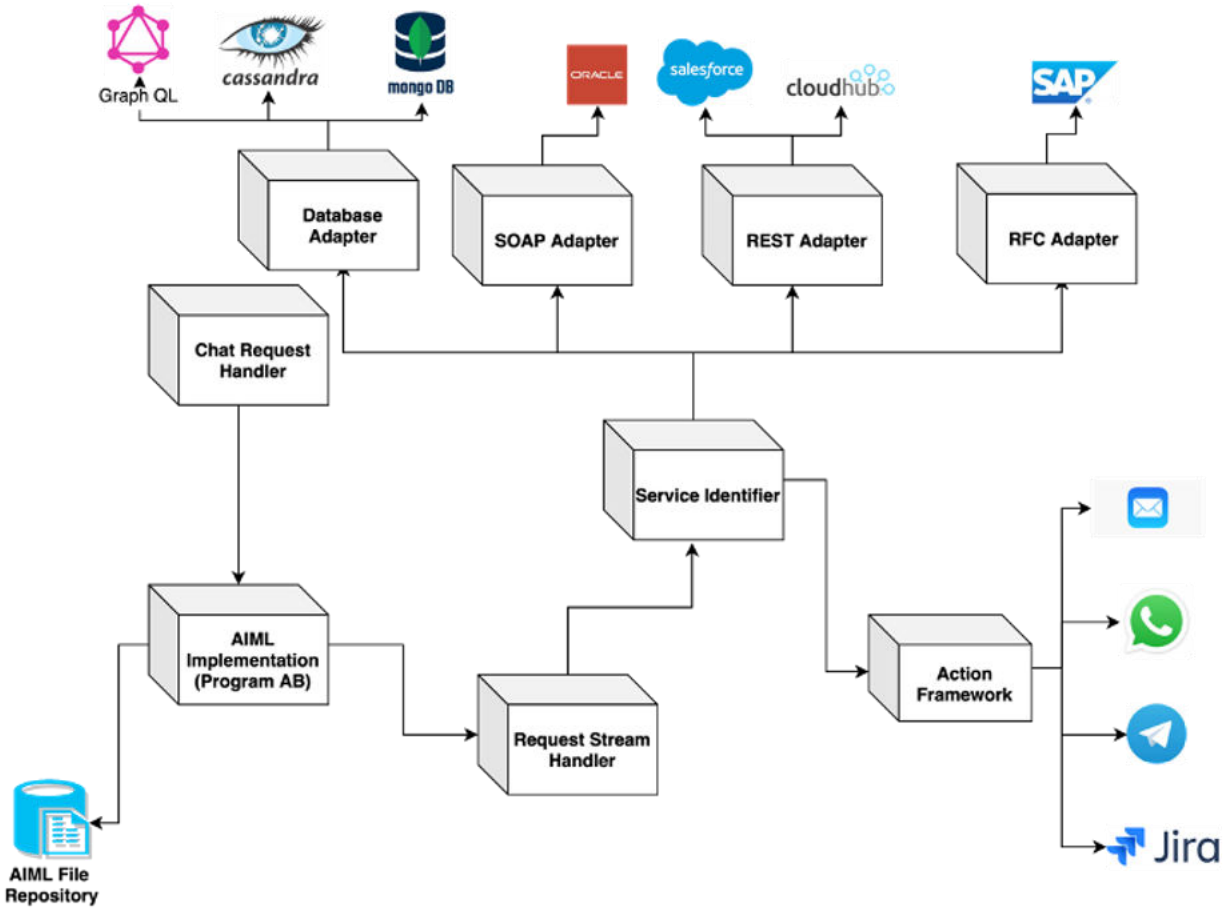

Figure 7:Proposed framework for Generic Service learner BOT

The whole functionality is wrapped up in the response handler API[15]. This hash map is used to fill Mother AIML file template and later it is loaded to the Bot at run time and the bot will give the best response from the available option. Which will be later shown to the end user at the chat-bot UI.

Proposed framework for Generic Service learner BOT:Chat-bot Application is made on top of Program AB, an Open Source reference Implementation in Java of AIML, the language commonly used to create conversational chat-bots. The Application takes the AIML templates used for basic User interactions and then builds its own custom AIMLs on top of it with the help of the REST Adapter discussed above. All these modules are integrated in the form of a Spring Boot application and the user interacts with the RAML Learner BOT using Angular UI.

Users can invoke the Bot by saying Hi or Hello Bot and ask the simple generic question like what the time is, how is the day etc and BOT will prepare the response using NLP and respond back to the user. If the query is complex, like to interact through the web- service the processing will take place with the help of AI and RAML. Figure 6 explains the flow while interacting through the infrastructure. For e.g.: when the user asks what is my CloudHub server status, the bot will first check if the user is already authenticated or not? If the communication is being established for the first time, it will ask for the credentials and store it in session attributes for future use. After getting the credentials, it will authenticate the user by validating username and password.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

CloudHub generates the authorization token which will be stored to maintain the session and the user will get the successful authentication response. After successful authentication, the user can ask any query related to CloudHub. Next time if the users ask any query related to CloudHub, using these tokens, BOT directly responds to those queries without calling the authorization flow.

Flow Diagram of proposed methodology for CloudHub POC:We can also interact through this BOT by giving voice commands. The most efficient technique to achieve this is by integrating this with Alexa. We have designed an Alexa skill that will call the CloudHub APIs internally and provide responses based on the utterances used in user query[16].

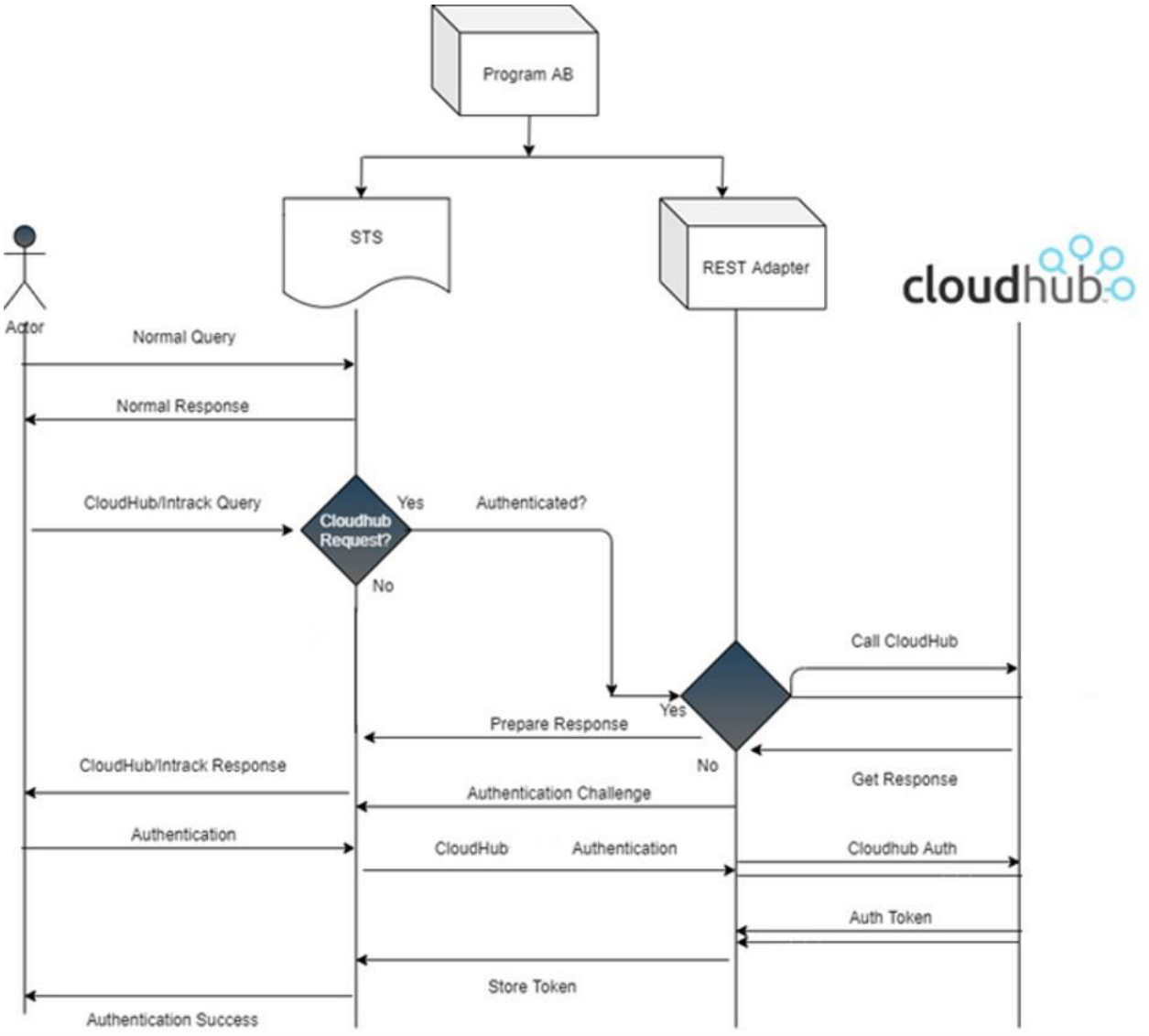

Figure 8:Flow Diagram of proposed methodology for CloudHub POC

IV. EXPERIMENTAL RESULTS

The aim of this experiment is to investigate the performance of the user intent identification of the bot and the context information extraction of the RAML parser. Once we have the AIML generated and intent is mapped, the generation of appropriate response to the user is straightforward.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

1. **Background**: There are two instances where classification comes into picture. One, during the RAML parser, and other during intent identification by chatbot.During the first classification, we are reading through the API endpoints and then generating the AIML files based on it. The failure is minimized by providing the parser with a valid and descriptive specification file for crawling.In the second classification, two points of failure can occur.One, if the user Intent is not mapped correctly, which can happen if the application is provided with insufficient query details from the User. Second, if, during the AIML generation of two different specs, two separate sets of RAML will be created. It might happen that those two spec have some similar API endpoints. This may create points of failure. However, if sufficient context is provided by the user as to which service the query is for, the response from the bot is very robust.

2. **Setup:**During the reading of API specification file, the best way to measure the efficiency is by creating a decision tree. We will calculate entropy for each class and determine the best possible Decision tree structure that can be created. This decision tree will then be used at the intent mapping stage to get the proper result to the user. The entropy is calculated using the formula:

$$E(S) = \sum_{i=1}^{c} - p_i log_2 p_i$$

where p_i is the probability of an element per class i in the data. We will use F-score to evaluate the performance of the application during intent recognition and mapping stage. The dataset is created manually, having 2386 queries in Training set and 735 in testing dataset.The F-score method computes the efficiency on two well known measures, which are precision and recall:

$$P_i = \frac{TP_i}{TP_i + FP_i} \qquad R_i = \frac{TP_i}{TP_i + FN_i} \qquad F_i = \frac{2.P_i.R_i}{P_i + R_i}$$

Where \$P$_i$ R$_i$ and F$_i$ are the precision, recall and F-score for the prediction of class i respectively. TP_i is the number of items correctly predicted in class i; FP_i is the number of items predicted incorrectly to class i; and FN_i is the number of items of class i that is mistakenly predicted into other classes.

3. **Experimental Results and Comments:**With weighted entropy of ≈ 0.861, the decision tree with base URL as the root node, doing Multi-Way split to get the tree and stopping when we have reached the HTTP requests was found to be the most efficient decision tree. The average results for the user intent identification are quite promising, with an average-F1 score of 95.33. The user intent identification gives a very high score($\approx 98.3\%$) for api and organization related queries(cpu size, name of api deployed, worker etc.). However, some endpoints, like /domain, due to multiple matching, sometimes gives erroneous responses to the user, essentially fetching the data from wrong AIML than it is supposed to due to ambiguity. Consequently, the score for these APIs goes down to $\approx 78.9\%$. One way to improve the scores we can adopt either invalidating the sessions and removing the references of APIs that are not in context. This solution will come into picture if the parser is crawling through multiple independent API specifications. The other way is to look at the context sensibility of the the conversation and running more mature algorithms that can improve contextual search. We can do that by asking end-users followup questions, to decrease ambiguity.

V. CONCLUSION AND FUTURE WORK

We have developed a RAML Learner BOT which is built on top of dynamic learning. It uses the combination of four techniques viz, NLP, AIML, ML on decision Trees and RAML. It has been developed using open source which does not require any cost. This intellectual BOT is generic and can be adaptable by most of the web services like CloudHub, SAP services etc. And is extendable over different data protocols like Database and CRM etc. which can expose APIs or otherwise. Its serverless deployment makes it low cost and less maintainability by offering greater scalability and much flexibility. In future, we can develop a custom crawler that has the capability to maintain the micro services as well. We can also train this model by using ML over logs like rsyslog or custom logs to provide solutions for a more efficient log analysis and troubleshooting process and pluggable escalation or relevant actions, while decreasing unnecessary human effort and increasing the accuracy of the mapped troubleshooting activities . The Actionable framework the implementation offers can plugin different

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 10, October 2022||

| DOI:10.15680/IJIRSET.2022.1110078|

communication channels like Telegram, WhatsApp, e-Mail or over workflows which can be triggered as part of escalation mechanisms. In future work other protocols and the actionable framework with log integration with ML and custom actions can be tried.[17]

REFERENCES

[1] . R. Bavaresco, D. Silveira, E. Reis, J. Barbosa, R. Righi, C. Costa, R. Antunes, M. Gomes, C. Gatti, M. Vanzin and others, "Conversational agents in business: A systematic literature review and future research directions," Computer Science Review, vol. 36, p. 100239, 2020.
[2] . M. McTear, "Conversational AI: dialogue systems, conversational agents, and chatbots," Synthesis Lectures on Human Language Technologies, vol. 13, p. 1–251, 2020.
[3] . B. A. Shawar and E. Atwell, "Chatbots: are they really useful?," in Ldv forum, 2007.
[4] . M. Daswani, K. Desai, M. Patel, R. Vani and M. Eirinaki, "CollegeBot: A Conversational AI Approach to Help Students Navigate College," in International Conference on Human-Computer Interaction, 2020.
[5] . P. Vanichvasin, "Chatbot Development as a Digital Learning Tool to Increase Students' Research Knowledge.," International Education Studies, vol. 14, p. 44–53, 2021.
[6] . C. S. Kulkarni, A. U. Bhavsar, S. R. Pingale and S. S. Kumbhar, "BANK CHAT BOT–an intelligent assistant system using NLP and machine learning," International Research Journal of Engineering and Technology, vol. 4, p. 2374–2377, 2017.
[7] . B. S. Prasad and C. M. V. S. Akana, "Polarity Sentiment-Based Intelligent Chat Bot For Judicious Customer Service Escalation," in 2021 2nd Global Conference for Advancement in Technology (GCAT), 2021.
[8] . M. Adam, M. Wessel and A. Benlian, "AI-based chatbots in customer service and their effects on user compliance," Electronic Markets, vol. 31, p. 427–445, 2021.
[9] . Y. Wei, X. Zhu, B. Sun and B. Sun, "Comparative studies of AIML," in 2016 3rd International Conference on Systems and Informatics (ICSAI), 2016.
[10]. N. Ashikhmin, G. Radchenko and A. Tchernykh, "RAML-based mock service generator for microservice applications testing," in Russian Supercomputing Days, 2017.
[11] . P. Canas, D. Griol and Z. Callejas, "Towards versatile conversations with data-driven dialog management and its integration in commercial platforms," Journal of Computational Science, vol. 55, p. 101443, 2021.
[12] . M. Gómez-Albarrán, A. Sarasa-Cabezuelo, J.-L. Sierra-Rodríguez and B. Temprado-Battad, "Authoring and playing interactive fiction with conventional web technologies," Multimedia Tools and Applications, p. 1–43, 2021.
[13] . E. Adamopoulou and L. Moussiades, "An overview of chatbot technology," in IFIP International Conference on Artificial Intelligence Applications and Innovations, 2020.
[14] . H. T. Hien, P.-N. Cuong, L. N. H. Nam, H. L. T. K. Nhung and L. D. Thang, "Intelligent assistants in higher-education environments: the FIT-EBot, a chatbot for administrative and learning support," in Proceedings of the ninth international symposium on information and communication technology, 2018.
[15] . V. Bajoria and A. Singh, Control your Mule Infra Via Chatbot, 2020.
[16] . Razia, Cloud Hub Alexa Skill Set - Command your IAAS to work for you, 2020.
[17] . A. Tiwari, Give Voice to Your Mule Message, 2020.
[18] . L. Maydwell, A Haskell RAML library on GitHub.
[19] . M. Alencar and J. Netto, "Developing a 3D conversation agent talking about online courses," in EdMedia+ Innovate Learning, 2011.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details