Octopus: A Multi Agent Chatbot

B Hettige# and AS Karunananda
Faculty of Computing, General Sir John Kotelawala Defence University, Ratmalana, Sri Lanka
#budditha@kdu.ac.lk

Abstract: *A Chatbot is a type of conversational agent or a computer program which has been designed to simulate an intelligent conversation with one or more human users via auditory or textual methods. Chatbots can be used as an exciting intelligent mechanism to interacting with computers. Undisputedly, Chatbots revolutionize the next generation of human-machine and machine-machine interaction in the context of Intelligent PC assistants. This paper presents design and implementation of multi agent based Sinhala Chatbot, named as Octopus. It consists of 8 sub multi agent systems namely core system, GUI system, Natural Language Processing system, communication system, learning system, action system, searching system and data access system to handle its intelligent capabilities. Octopus has been implemented through Java and capable to run with Windows and Linux. The Octopus has been incrementally tested and has shown encouraging results in its intelligent performance.*

Keywords*— Multi-Agent System, MaSMT, Chatbot*

I. INTRODUCTION

In 1956, the field of Artificial Intelligence was born with a primary objective to develop intelligent machines. The emergence of the field of AI was very much influenced by the classic article by Alan Turing (Turing, 1950). His article has presented how the machine intelligence could be evaluated through what is called Turing Test. Since mind 1950s, people have been struggling to develop an intelligent machine that could pass the Turing test. In such attempts, special software known as Chatbot has shown reasonable capacity to pass the Turing Test with a high level accuracy.

A chatbot is a type of conversational agent or a computer program which has been designed to simulate an intelligent conversation with one or more human users via auditory or textual methods (Imran 2015). Computer-based natural language processing has been seen as the key intelligent feature of a Chatbot. Further, chatbot provides interface between computer and user with intelligent features. At present, numberof chatbot systems are available on the web including Cleverbot (Cleverbot, 2015), ALICE (ALICE, 2015), Mitsuku (Mitsuku, 2015), Eliza (Eliza 1966) and Sinhala chatbot (Hettige and Karunananda, 2006).

Some of the above chat systems (including ALICE and Cleverbot) provided intelligent behaviour and passed the Turing's test during few years. However, development of the intelligent chatbot has been coincided as a research challenging task in the area of Natural Language Processing (NLP).

According to the complexity of the natural languages, Implementation of the chatbot system is not an easy task. It requires natural language understanding, Natural language generation and a mechanism to knowledge extraction. There are number of tools available for chatbot development including AIML (Marietto, 2013). Artificial Intelligence Mark-up Language (AIML) is an XML dialect for creating natural language software agents. The AIML was developed by Richard Wallace with the ALICE chatbot and released under the GNU GPL.

Multi agent system (MAS) technology is a modern software technology that can be used to handle complexity (Rzevskiand Skobelev, 2014). In addition to the above, multi agent systems have some built-in features including autonomy, pro-activity and social ability. These features can be easily used to enhance intelligent performances of the chatbot systems. However, Chatbot system development through the multi agent technology is a novel approach for the area of Natural Language Processing.

This paper presents an improved version of the existing Sinhala chatbot, name Octopus, designed through the multi agent architecture and implemented using multi agent development framework MaSMT (Hettige et.al. 2013). The Octopus consists of 8 sub multi agent systems namely core system, GUI system, Natural Language Processing system, communication system, learning system, action system, searching system and data access system.

Initially Octopus has been design to provide chat facilities through text. This version also provides action and searching facilities to enhance the performance of the Octopus. The action facilities of the Octopus handles limited tasks including execute some command, open or close some application search some result etc. The Octopus has an ability to search some files or data inside the PC or throughout the local network.

The rest of this paper is organized as follows. Section 2 describes overview of some existing chatbot systems including their intelligent behaviours. Section 3 reports brief note on Multi agent system technology including MaSMT framework. Section 4 also gives design and implementation of the Octopus. Then section 5 demonstrates how octopus in action. Finally section 6 concludes the paper with a note on further work.

II. INTELLIGENT CHATBOTS

The Turing test is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human (Turing, 1950). ELIZA is the first program created by Joseph Weizenbaum which appeared to pass the Turing test (Weizenbaum, 1966). In addition to the above some other chatbot systems already passed the Turing test including ALICE and Cleverbot. Most of these chatbot systems demonstrate some intelligent features through text or voice output. To demonstrate intelligent features most of the chat systems use AIML knowledge base. The AIML knowledge base consists of data objects (AIML objects) and categories. Category is a rule for matching an input and converting to an output. The following code gives a sample AIML document.

```
<aiml version="1.0.1" encoding="UTF-8"?>
<category>
     <pattern> HELLO BOT </pattern>
<template>
      Hello my new friend!
</template>
</category>
</aiml>
```

The basic units of an AIML dialog are called categories. Each category is a fundamental unit of knowledge contained in the Chabot's Knowledge base. A category consists of a user input, a response to user input, presented and an optional context about the category. The <pattern> tag contains a possible user input. The <template> tag contains possible chatbot answers to the user. It must be within the scope of a <category> tag. Most of the chatbot systems use AIML to build their Chatbots' knowledgebase. Some selected chatbot systems and their features discuss in the below.

A. ELIZA

ELIZA is computers program that early example of a primitive natural language processing. ELIZA passed the Turing test in 1966. The ELIZA program first implemented through SLIP with simple parsing methods. A basic version of ELIZA appeared in Creative Computing in 1977.

B. ALICE

ALICE (Artificial Linguistic Internet Computer Entity) is a popular award winning Open source, free Artificial Intelligence chat robot that was originally composed by Richard Wallace in 1995. Alice uses AIML to specifying the heuristic conversation rules(Wallace, 2009). Further, ALICE system won the Loebner Prize in year 2000, 2001, and 2004. Note that, the Loebner Prize is the oldest Turing Test contest, started in 1991 by Hugh Loebner and the Cambridge Centre for Behavioural studies (aisb, 2010). The Alice system consists of two modules namely Alicebot engine and AIML knowledge bases. The Alice system can be downloaded from Alice web site.

C. CleverBot

Cleverbot is an Intelligent bot system that can carry on a conversation as well as any human (Cleverbot, 2015). The Cleverbot has been implemented through a web application including artificial intelligence algorithm to provide conversations with humans. Compare with other chat systems Cleverbot comes with some special features. For instance, response of the Cleverbot is not programmed. The Cleverbot system reads the human input and finds all keywords or an exact phrase. After searching through its saved conversations, it responds to the input by finding how a human responded to that input.

D. Mitsuku

Mitusuku is also an intelligent chatbot that was created through the Alice's AIML files (Mitusuku 2012). Mituku includes the ability to reason with specific objects. For instance, if someone says "Can you eat a house?" Mitsuku looks up the properties for "house". Finds the value of "made_from" is set to "brick" and replies no, as a house isn't edible.

E. Sinhala chatbot

A Sinhala chatbot (Hettige and Karunananda 2006) has already developed by using Prolog based Natural

Language processing modules such as Sinhala parser (Hettige and Karunananda 2006a) and Sinhala Morphological analyser (Hettige and Karunananda 2006b). These Sinhala language processing modules have been developed under the BEES project (Hettige 2011). The entire chatbot system (Sinhala chatbot) has been developed using JAVA and SWI-PROLOG that runs on both Linux and Windows. Figure 1 shows the design diagram of the existing Sinhala chatbot.

The Sinhala chatbot consists of 4 modules namely core system, Knowledge identification engine, knowledge base and application module. The core system of the Sinhala chatbot consists of 4 Natural language processing modules namely Sinhala Morphological analyser, Sinhala parser, and Sinhala morphological generator and Sinhala sentence composer.

The Sinhala Morphological analyser reads an input sentence word by word and identifies the morphological information of the Sinhala word such as noun, verb etc. then Sinhala parser analyses the input sentence (Text) and identify the structure of the Sinhala text. These two modules are used to identify given text (Natural Language Understanding). Then Knowledge identification module search appropriate text to generate the Sinhala output. Sinhala chatbot also has an ability to get support from action module to do some actions (Open/close some application, executes some command etc.,) After that Sinhala composer compose the Sinhala text and Sinhala morphological generator generates appropriate Sinhala words according to the Sinhala grammar. All the above Language supporting modules have been developed through the SWI-Prolog (swi-prolog, 2010). The main purpose of the Sinhala chat system is to demonstrate the capabilities of the Prolog based Natural language processing modules such as Sinhala parser, Sinhala composer, Sinhala Morphological analyser and Sinhala morphological generator.

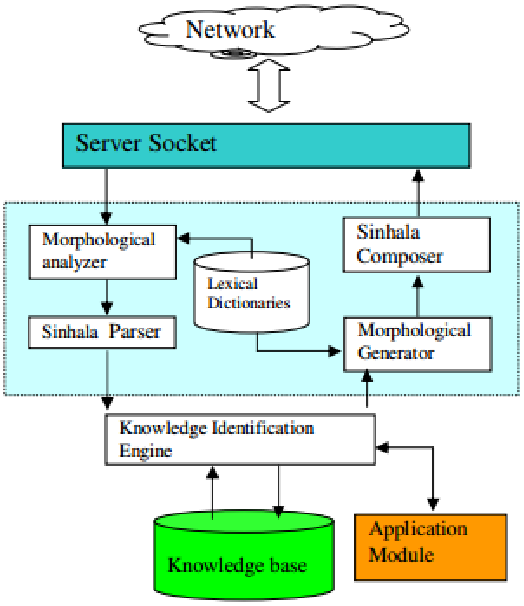


Figure 1: Overview of the existing Sinhala chatbot

III. MULTI AGENT SYSTEMS

Octopus has been designed and developed through the multi agent technology. This section briefly describes multi-agent technology and its usage. The Multi agent system explores new software paradigms to model complex systems. Most of these multi agent systems are large networks of small agents which are run in parallel (Rzevski 2014). Note that, performance of the Multi agent system depends on how agents are designed and capability of the agent negotiations. Therefore, different approaches are available to model agents including common standards. Further, numbers of agent development frameworks (Jade, MadKit, MasMTetc) supported to develop Multi-agent systems with agent development standard such as FIPA-ACL and KQML. Some existing multi agent system development framework including MaSMT is briefly described in the below.

JADE (Java Agent Development Framework) is a software framework fully implemented in Java language. The JADE framework provides supporting GUI tools for debugging and deployment phases in multi agent developments.

The AgentBuilder is an integrated software development tool that allows software developers to build agents quickly and easily without sound knowledge of Multi -agent technology.

SeSAm (Shell for Simulated Agent Systems) is another framework that provides a generic environment for multi agent system development. Compare with others MadKit is also multi agent system development tool that provides lightweight Java library to design and simulate multi agent systems.

MaSMT (Multi agent System for Machine Translation) is a Java based Multi agent system development framework, which is especially design to develop English to Sinhala machine translation system. Figure 2 shows the design architecture of the MaSMT framework.

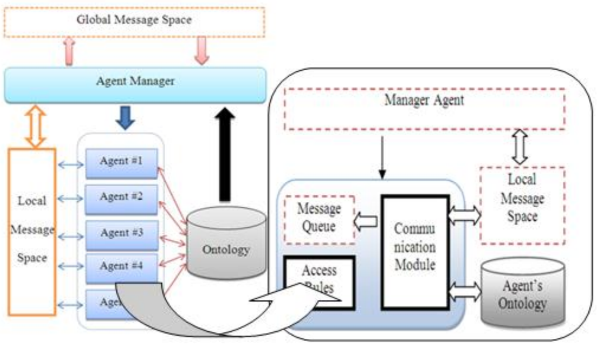


Figure 2: design architecture of the MaSMT framework

The MaSMT framework originally comprises of two types of agents, namely ordinary agent and manager agent. Further, manager agent can directly communicate with other manager agents and each and every ordinary agent in the swarm is assigned to a particular manager agent. An ordinary agent in a swarm can directly communicate only with the agents in its own swarm and its manager agent. The MaSMT framework primarily implements object-object communication, XML based data passing and MySQL database connectivity with message passing methods. Agent communication in the framework has been implemented to comply with FIFA -ACL specification. Improved version of the MasMT can be downloaded from the sourceforge web site (sourceforge, 2015).

IV.ANATOMY OF OCTOPUS

This section briefly discusses design and implantation of the Octopus. Octopus is a multi-agent system that consists of 8 sub multi agent systems namely core system, GUI system, Natural Language Processing system, communication system, learning system, action system, searching system and data access system. These 8 subsystems can independently work and coordinate by the core system. The Octopus (chatbot) has been designed through the MaSMT framework. Figure 3 shows the anatomy of the Octopus. The following sections give brief description of the each sub systems.

A. The core system

The core system coordinates the other 7 sub multi agent systems. The core system has been designed as a manager. Therefore, core agent can manage the other agent as required. The core system has special features. The core agent can communicate with other Octopus which is running on the same network. Further, octopus can directly commutate with its mother system or other system on the web and get required information/updates.

B. The GUI System

The GUI system handles user input and output of the Octopus. The GUI system consists with a manager agent and 2ordinary agents namely text agent and voice agent. The text agent is a graphical user interface that provides text input method and status control mechanism of the octopus. By using this interface user can manually control the Octopus. The text agent also provides default input and output interface to enter or display text. In addition to the normal input octopus has been capable to provide voice output for the English text through the voice agent. The voice agent has been designed with java based freeTTS (Text to speech) modules.

C. The Ontology

Each agent of the system has its own shared ontology. Initially Ontology of the octopus has some limits. However, Octopus system can update its own ontology while you are using the system. To update the existing ontology system get support from Sinhala Ontological generator(Hettige and et.al 2015) that was developed to generate word ontology in Machine Translation (Hettige and et.al 2014).Octopus's ontology has been design through the AIML.

The ontology of the system can be divided in to 4 categories such as user ontology, agent ontology, language specific ontology and knowledge ontology. The user ontology gives information about the particular user profile and data. Agent ontology provides ontology about agents. This agent ontology consists of agent specific rules and agent knowledge that are required to execute agent task on the system. The language specific ontology is the Natural Language data (Sinhala and English) that are required to communicate between user and the PC. The Knowledge ontology is a dynamic ontology that may be change/update when the system running. For instance user asks the meaning of a particular word then Octopus searches it (word) on its knowledge based. If it is not available on its ontology, then system capable to search this data through the internet by using search agent.

D. *The Natural Language Processing system*

The Natural Language processing system comprises with a manager agent and number of natural language processing agents including natural language analysis and natural language generation. These agents handle all the user inputs and provide the output as required. Note that, octopus system does not give much attention on both morphology and syntax of the input text. It also assumes that, users are not fluent in both Sinhala and English languages. However, system gives more attention on semantics on the text.

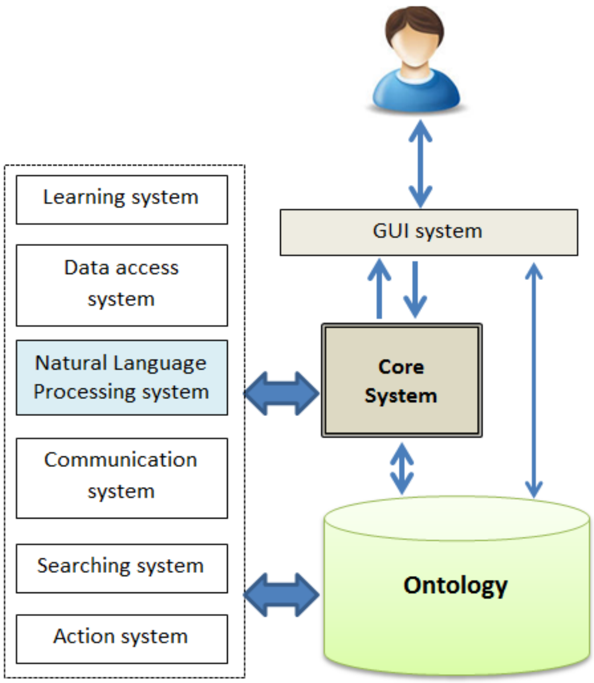


Figure 3: Design of the Octopus

E. *The Communication system*

The communication system handles all the required communication (Message parsing) of the Octopus including agent-agent, agent-system and system-system communications. The agent-agent communication has been done through the XML source and system-system communication has been done through the client-server networking. The system also capable to communicate with MySQL databases to collect data from a database.

F. *The Learning system*

The learning system comprises with a manager agent and number of learning agents to study and update the system. The learning agents are capable to read and update the relevant knowledge ontology while running the system. To enhance the intelligent of the octopus the learning agent handles the most important role. The leaning agent also has ability to update other agent rules as well as knowledge base.

G. *The Action system*

The action system comprises with a manager agent and number of action agents to do some actions. The action agent is used to execute relevant tasks and it also helps to reduce the user work load. This agent can handle two types of actions namely user required actions and automated actions. The user required action is the action that user requested. For Instance, octopus can work as a user (assume) of the computer and do some actions such as open some programs, close some programs read some files and execute some command through the command prompt etc. Automated action is a type of action done by the octopus automatically including copy files into some places, search some files, execute some command etc.

H. *The Searching system*

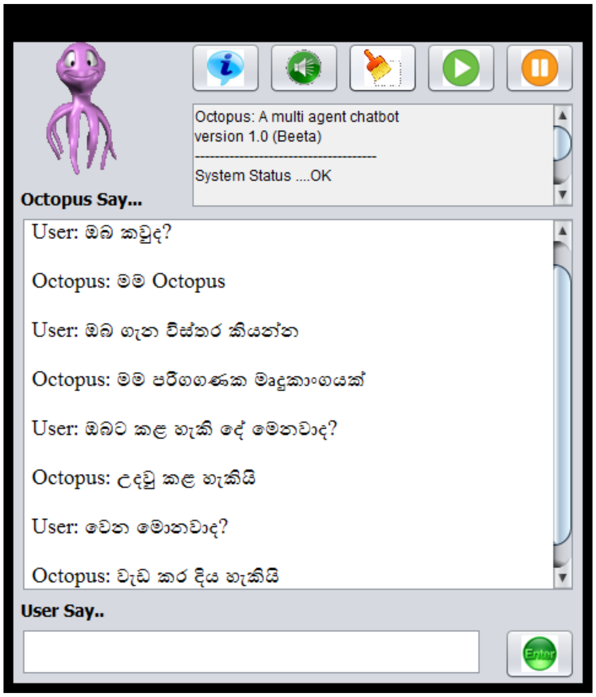
The searching system comprises with a manager agent and number of searching agents. The search agents can search information from internet or its own network. The internet search agent uses JSOUP library (Jsoup, 2009) to collect data from web resources. For an instance, user can ask a meaning of a particular word Octopus can search result from web resources and collect the result. In addition to the above, searching agents are also capable to update the knowledge base as required. This behaviour is quite smiler to the behaviour of the Mitsuku chatbot. However, at present Octopus has very less number of capabilities on that action.

I. *Data Access system*

The data access system comprises with a manager agent and number of data access agents. These agents are used to copy required data trough out the PC or send or reserve data from another octopus systems. The Octopus uses client-server networking to access data throughout network.

V. OCTOPUS IN ACTION

This section briefly describes how octopus in action. The Octopus system has been implemented as a Java application and capable to run on any environment such as Windows and Linux. Figure 4 shows the User interface of the Octopus.



Octopus: A multi agent chatbot
version 1.0 (Beeta)

System Status ....OK

User: ඔබ කවුද?

Octopus: මම Octopus

User: ඔබ ගැන විස්තර කියන්න

Octopus: මම පරිගණක මෘදුකාංගයක්

User: ඔබට කළ හැකි දේ මෙනවාද?

Octopus: උදවු කළ හැකියි

User: වෙන මොනවාද?

Octopus: වැඩ කර දිය හැකියි

Figure 4: User Interface of the Octopus

Initially Octopus consists with very limited knowledge base. At the initial stage it uses system properties and collects required user information and make a user profile for the particular user by using AIML. In addition to the above system communicate with other octopus that are available in the current network and get required knowledge or user profile if it is exits. For Instance, if another profile available on the network with the same name system should ask to get it and collect it. After that, Octopus system ready to work. User can communicate with octopus through the input text. After collecting the input from user, GUI system sends it to the message space. The message space is the common area of the multi agent system that used to share the task and knowledge. Then Natural language processing system tries to identify above text on knowledge base if it is exit then provide the output. If it is directly not exit on the knowledge base searching system tries to find the solution from network or web. After collecting some solution it send it to the message space. Finally GUI agents display it as a text output of voice. Further, user requires some action than the text output, the action system collect the required information and tries to execute the action.

VI. CONCLUSIONS AND FURTHER WORKS

This paper presented design and implementation of the improved Sinhala chatbot name Octopus. The Octopus has been design as a multi agent system and implemented through the multi agent system development framework MaSMT. The Octopus consists of 8 sub-systems namely core system, GUI system, Natural Language Processing system, communication system, learning system, action system, searching system and data access system. The Octopus has been implanted as a Java application and successfully tested with the laboratory environment.

This version of the Octopus consists of limited capabilities of the semantics processing. Therefore, update Language processing sub system is essential to enhance the system intelligent. Finally, pass the Turing's test can be considered as a further work of the research.

REFERENCES

ALICE (2010), <http://alice.pandorabots.com> Accessed 25 May 2015.

Cleverbot (2015), <http://www.cleverbot.com/> Accessed 25 May 2015.

Eliza (2010), <http://www.manifestation.com> Accessed 02 June 2015.

Hettige, B., Karunananda, AS., (2006). First Sinhala Chatbot in action, in: Proceedings of the 3rd Annual Session, Sri Lanka Association for Artificial Intelligence (SLAAI), Colombo, 4–10.

Hettige, B., Karunananda, A.S., Rzevski, G., (2013). MaSMT: A Multi-agent System Development Framework for English-Sinhala Machine Translation. Int. J. Comput. Linguist. Nat. Lang. Process. IJCLNLP 2, 411–416.

Hettige B., Karunananda AS, Rzevski G., (2014) A Sinhala Ontology Generator for English to Sinhala Machine Translation, Proceedings of KDU International Research Symposium.

Hettige B., Karunananda AS, Rzevski G.,(2015), Multi-agent solution for managing complexity in English to Sinhala Machine Translation, Proceedings of International

Conference on Complex Systems in Business, Administration, Science and Engineering, New Forest, UK.

Imran A., Shikha S.,(2015), AIML Based Voice Enabled Artificial Intelligent Chatterbot, International Journal of u- and e-Service, Science and Technology, Vol.8, No.2, pp.375-384.

Jsoup (2009), <http://www.jsoup.org>Accessed 05 June 2015.

Mitsuku (2015), <http://http://mitsuku.com/> Accessed 05 June 2015.

MadKitc (2010), <http://www.madkit.net> Accessed 05 June 2015.

Marietto, MDGB., et. al. (2013), Artificial Intelligence Markup Language: A Brief Tutorial, International Journal of Computer Science & Engineering Survey.

Rzevski G., Skobelev P., (2014), Managing Complexity, WIT Press.

Sourceforge (2015) <http://sourceforge.net/projects/masmt> Accessed 05 June 2015.

Swi-prolog (2012), <https://www.swi-prolog.org> Accessed 05 June 2015.

Society for the Study of Artificial Intelligence and Simulation of Behaviour, (1995), <http://www.aisb.org.uk/>Accessed 03 June 2015.

Turing, A.M., (1950). Computing machinery and intelligence. Mind 433–460.

Wikipedia (2012), <https://en.wikipedia.org> Accessed 05 June 2015.

Weizenbaum J., (1966), "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine", Communications of the ACM 9 (1): 36–45.

BIOGRAPHY OF AUTHORS

B Hettige is a PhD student of the Faculty of Information technology, University of Moratuwa, Sri Lanka. At present he is a probationary lecturer of the department of Computer Science, faculty of Computing, General Sir John Kotelawala Defence University. His research interests include Multi-agent technology, Natural Language processing and Sinhala Computing. He has produced more than 25 referred international and local publications to his credit.

AS Karunananda is a Senior Professor of Information Technology University of Moratuwa, Sri Lanka. At present he is the Dean of Research and Development and Faculty of Computing of General Sir John Kotelawala Defence University. His research interests include Multi Agent Systems, Ontological Modelling, Machine Translation, and Theory of Computing.