AN ARCHITECTURE DESIGN OF THE INTELLIGENT AGENT FOR SPEECH RECOGNITION AND TRANSLATION

Abd. Manan Bin Ahmad, Ag.Noorajis bin Ag.Nordin,
Emrul Hamide Md. Saaim, Den Fairol bin Samaon, Mohd Danial Bin Ibrahim

Speech Recognition Laboratory,
Department of Software Engineering,
Faculty of Computer Science and Information System,
University of Technology Malaysia

ABSTRACT

An architecture design of the intelligent agent for speech recognition and translation is presented in this paper. The design involves the agent architecture and the method of the agent is used . The architecture design will show the relationship between the intelligent agent and speech recognition also translation. The intelligent agent for speech recognition is called S – AGENT and T – AGENT f or translation. The purpose of the S – AGENT is to facilitate for transmitting the speech data via internet or network. The S – AGENT is acting as a data transmit control to ensure the transmitted speech data will secure delivered. The task of the T – AGENT is different from the S – AGENT. The T – AGENT is acting as information retrieval. It will process the output from the speech recognition and translates the output based on its information memory database. If the information cannot be found on its memory, it will search the information required from the database dictionary provided. At the same time, it learns the information and saves the information to its memory for the future purpose.

1. INTRODUCTION

As wide-spread open network such as internet and intranets grow larger, demand for distributed systems with rapid changes and diverse requirements is getting higher. Agent technology is expected to be a basis for the development of such systems. Agents are software units that can deal with environmental changes and diverse requirements with their features such as autonomy, mobility, intelligence, cooperation and reactivity [1]. At the same time, speech researchers and computer software engineers have been putting a great deal effort to integrate speech functions into Internet applications [2]. They are trying to design a distributed speech recognition system using internet applications.

This paper presents an architecture design of the intelligent agent for speech recognition and translation system via internet or network. The design is divided in two parts; first part: an architecture agent design of speech recognition is and second part: an architecture agent design of translation. Each agent has own task, responsibility and capability. The basic flow of the speech recognition and translation system using the intelligent agent is shown in figure 1.

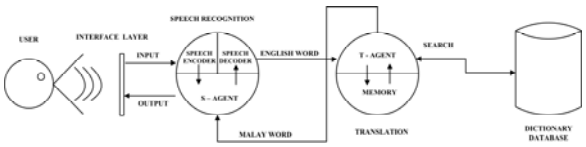

Figure 1. The flow diagram for speech recognition and translation system using the intelligent agent.

The user will speak a word in English and record the voice into the interface layer. The interface layer is a web-based application (*html*) that can receive and record voice from the user. After the voice is recorded, it will send to the speech encoder. The speech encoder processes the input and extracts the features of the voice. The extract data send to the *S – Agent*. The *S - Agent* will send the extract data to the speech decoder when the network is in good conditions. The speech decoder will recognize the extract data and covert it into a text. The text will send to the *T – Agent* for translation purpose. The *T – Agent* will search the related information from the dictionary database when the text information was not found in its memory. The information required (in Malay word translation) will send back to the *S – Agent*. The *S – Agent* will give the required information to the user.

From the figure 1, the concept of distributed speech recognition is also implemented. The speech recognition has been divided into several processes such as speech encoder (feature extraction) and speech decoder (speech recognizer). Overall, the system will receive input in the form of spoken words and enable the system to search the database and return exact or best fitting result of the search.

Several concepts and characteristics of an agent will be applied on the design. For this architecture design, the concept of mobile agents and adaptive agents will be used because of their advantages. The details of the architecture design will explain in section 2. In section 3, the conclusion is included in this paper.

0-7803-8560-8/04/$20.00©2004IEEE

2. THE ARCHITECTURE DESIGN

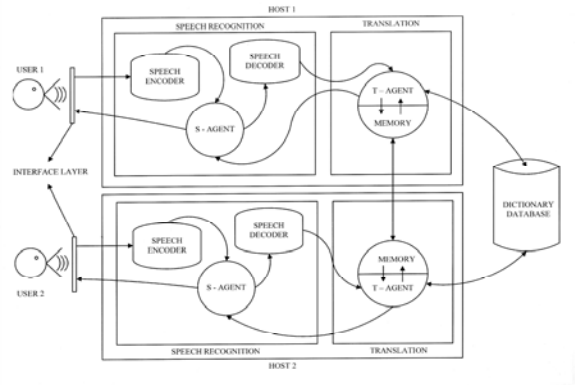

Figure 2. Speech Recognition and Translation Agent Architecture Design.

The figure 2 shows the high - level of the Speech Recognition and Translation Agent Architecture Design. The users will record theirs voice using the interface layer. The interface layer is a web application (HTML). The basic task for the interface layer is record user voice and sends to the speech decoder. After the users recorded theirs voice, they can submit their voice from the interface layer. The speech recognition will be separated into three part, the speech encoder, speech decoder and the *S – Agent*. The responsibility of speech encoder is to extract data from voice recorded. The speech encoder contains feature extractions, and acoustic processing. The *S – Agent* is a agent that facilitate for transmitting the speech data via internet or network. The S – AGENT is acting as a data transmit control to ensure the transmitted speech data will secure delivered. The speech decoder contains a speech recognizer. The output from the speech decoder is a text in English word and will send to the *T – Agent*. The details about the S – *Agent* will describe in section 2.1. The Translation has the *T – Agent*. *The T – Agent* will translate the word from English to Malay and save all the word on its memory for future purpose. The dictionary database is a dictionary that contains all word in English and Malay for translating purpose. The *T – Agent* Architecture Design will explain in section 2.2.

2.1. The *S – Agent* Architecture Design

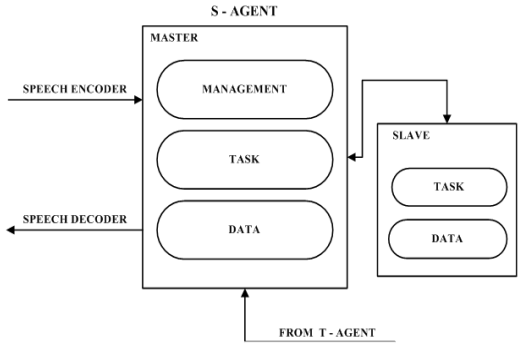

Figure 3. The *S – Agent* Architecture Design.

The *S – Agent* architecture design is shown in figure 3. The S – *Agent* has two parts; master and slave. The *S – Agent* master contains management, task and data. The *S – Agent* slave consists of task and data. The table 1 shows the basic role for each component in the *S – Agent*.

Management	This component will manage the slave activity and monitor the slave.
Task	For the master , the task has two process; create a slave and get result from the slave. For the slave, the task will get and send the data according from the master instructions.
Data	Data contains the output from the speech encoder and the speech decoder. If the slave is destroyed, the master will hold the data.

Table 1: Components rule for the *S – Agent*.

The *S – Agent* handles the incoming data and also, the sending data to the destination. It will control all the data transfer to ensure that the data will transmit safely and secure. The concept of the *S – Agent* is similar with the mobile agents. The pattern for the *S – Agent* is called master – slave pattern (or task pattern)

Mobile agents are software processes which can autonomously migrate from one host to another during their execution [10]. In other words, the agent can suspend its execution, migrate to another machine, and then resume execution on the new machine from the point at which it left off [12]. While roaming the Internet or visiting other machines, they do some useful work on behalf of their owners or originators [10]. Most importantly, an agent can choose different migrating strategies depending on its task and the current network condition change. Complex, efficient and robust behaviours are realized with surprisingly little code [12]. The use of mobile agents for distributed applications has several potential benefits: [10]

- *Asynchronous task execution:* while the agent acts on behalf of the client on a remote site, the client may perform other task.
- *More dynamic:* It is not necessary to install a specific procedure at a server before-hand and to anticipate specific service request types; a client or a service provider may send different types of agents to a server without the need to reconfigure the server.
- *Reduced communication bandwidth*: If vast amounts of server data have to be processed and if only a few relevant pieces of information have to be filtered out, it is more economical to transfer the computation to the data than to ship the data to the computation.

The strength of mobile agents also allow traditional clients and servers to offload work to each other, and

change who offloads to whom according to the capabilities and current loads of the client, server and network. Similarly, mobile agents allow an application to dynamically deploy its components to arbitrary network sites, and tore-deploy those components in response to changing network conditions [12].

The master – slave pattern allows a master agent can delegate a task to a slave agent. The slave agent will move to a destination host, perform the assigned task, and return with the possible result of the task [15]. There are several reasons why agents (termed masters) would like to create other agents (termed slaves) and delegate tasks to them. A master agent can continue to perform other task in parallel with the slave agent. Consider an agent – based application that provides a GUI (Graphical User Interface) for inputting data and displaying the intermediate results of a specific task to be performed remotely. With a single agent to provide the GUI and perform that task, it will not be possible to maintain the GUI after the agent has travelled from its origin to a remote destination. Alternatively, a stationary (immobile) master agent can provide and maintain a GUI while a slave agent moves to another destination, performs the assigned task, sends intermediate results, and finally returns and deliver the task's result to the master agent, which display it to client [15].

2.2. The *T – Agent* Architecture Design

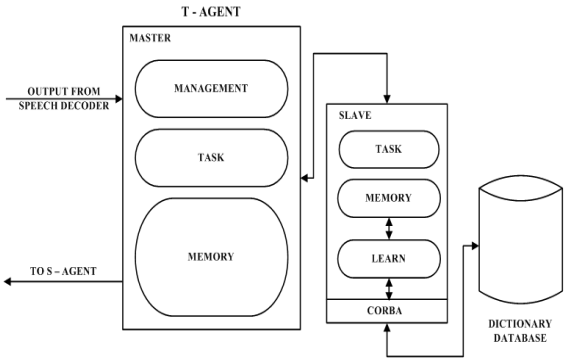

Figure 4. The *T – Agent* Architecture Design.

The *T – Agent* is an adaptive agent. The adaptive agent is combination of mobile agents and learning agents. The purpose of learning agents is to learn the information gather and develop the knowledge based for the agents. The agents must able to learn in order to improve their performance and adapt to the changes [7]. The pattern design for the T – Agent is similar with the *S – Agent*. However, the T - AGENT functions are different. The *T – Agent* receives input from speech decoder in English word text format. After the translation process is done, the word translate in Malay will send to the *S – Agent*.

The figure 4 shows the *T – Agent* Architecture Design. The *T – Agent* is divided into two parts. The *T – Agent* master is responsible to manage, create a slave and store the all information in its memory from the slave. The *T – Agent* slave will translate the input text from its memory. If the information about the input text is not found in its memory, it will search the related information from the dictionary database and learn the information and store the information into its memory. After the slave is finished its job, the slave add the new information learned into the master memory. The CORBA layer is a medium for communicating with the database.

3. CONCLUSIONS

An architecture design of the intelligent agent for speech recognition and translation had been discussed on this paper. The design involves the high - level architecture design, the *S – Agent* architecture and the *T – Agent* architecture. The design will be used for the implementation framework purpose. The concept of mobile agents and adaptive agents is applied in this design. The design will give a good idea how to implement the intelligent agent in speech recognition and translation.

4. REFERENCES

[1] Yasuki Tahara, Akikhiko Ohsuga and Shinichi Honiden, "Agent System Development Method Based On Agent Patterns," *In Proceeding ICSE '99*, ACM Press, USA, 1999.

[2] Z. Tu and P. Loizou, "Speech recognition over the Internet using Java," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 4, pp. 2367-2370, 1999.

[3] Campbell, M., J. Cagan and K. Kotovsky, "A-Design: Theory and Implementation of an Adaptive, Agent-Based Method of Conceptual Design," *Artificial Intelligence in Design '98 (J.S. Gero and F. Sudweeks, eds)*, Kluwer Academic Publishers, Dordrecht, Lisbon, July 20-23, pp. 579-598,1998.

[4] Pereira, F. B. and Costa, E., "How Adaptive Agents Learn to Deal with Incomplete Queries in Distributed Information Environments", *Proceedings of the Congress on Evolutionary Computation (CEC-2000)*, p. 1329-1336, 2000.

[5] B. Emako, R.H. Glitho and S. Pierre, "A Mobile Agent based Advanced Service Architecture for Wireless Internet Telephony: Design, Implementation and Evaluation," *IEEE Transactions on Computers*, Vol. 52, NO. 6, June 2003.

[6] C. Bernon, M-P. Gleizes, S. Peyruqueou, G. Picard, "ADELFE: A methodology for adaptive multi-agent systems engineering," *Paolo Petta Robert Tolksdorf Franco Zambonelli (szerk.): Proceedings of the 3rd International Workshop on Engineering Societies in the Agents World III (ESAW'03)*, 156-169. o., Springer Verlag, 2002.

[7] Fan Y, Gauch S., "Adaptive agents for

information gathering from multiple, distributed information sources," *Proceedings of 1999 AAAI Symposium on Intelligent Agents in Cyberspace*, Stanford University, pp. 40-46, March 1999.

[8] Daniela Rus, Robert Gray, David Kotz, "Autonomous and Adaptive Agents that Gather Information," *AAAI '96 International Workshop on Intelligent Adaptive Agents*, 1996.

[9] Lingnau A., Drobnik O., Domel P., "An HTTP-based Infrastructure for Mobile Agents," *WWW Journal - 4th International Conference Proceedings*, Boston, MA, Dec 11-14, 1995.

[10] Stefan Fünfrocken, Friedemann Mattern, "Mobile Agents as an Architectural Concept for Internet-based Distributed Applications - The WASP Project Approach," *In: Steinmetz (Ed.): Proc. KiVS'99*, pp. 32-43, Springer-Verlag, 1999.

[11] Heikki Helin, Heimo Laamanen, Kimmo Raatikainen, "Mobile Agent Communication in Wireless Networks, *European Wireless'99/ITG'99*. pp. 211-216, October 1999.

[12] Brian Brewington and Robert Gray and Katsuhiro Moizumi and David Kotz and George Cybenko and Daniela Rus, "Mobile Agents for Distributed Information Retrieval," *In Matthias Klusch, editor, Intelligent Information Agents*, chapter 15, Springer-Verlag, 1999.

[13] Bieszczad, A. and Pagurek, B., "Network Management Application-Oriented Taxonomy of Mobile Code," *Proc. of the EEE/IFIP Network Operations and Management Symposium (NOMS '98),* New Orleans, Louisiana, Feb. 15-20, 1998.

[14] Illmann, Krüger, Kargl, Weber, "Migration in Java: Problems, Classification and Solutions," *Proceedings of the MAMA'00*, Wollogong, Australia,December 2000.

[15] Yariv Aridor, Danny B. Lange, "Agent Design Patterns: Elements of Agent Application Design," *In Proceedings of Autonomous Agents '98*, ACM Press, USA, 1998.

[16] R. H. Glitho, E. Olougouna and S. Pierre, "Mobile Agents and Their Use for Information Retrieval: A Brief Overview and an Elaborate Case Study," *IEEE Network Magazine*, Vol.16 No1, pp. 34-4, January/February 2002.

[17] David Kotz and Robert S. Gray, "Mobile Agents and the Future of the Internet," ACM *Operating Systems Review*, 33(3):7-13, August, 1999.