

The Design of Interactive Conversation Agents

ONG SING GOH

Faculty of Information Technology and Communication
University Technical Malaysia Melaka, 75450
MALAYSIA
osgoh@ieee.org

CHUNG CHE FUNG

School of Information Technology
Murdoch University, Western Australia 6150
AUSTRALIA
l.fung@murdoch.edu.au

Abstract: - Interactive conversation agents or CAs are computer programs or application software designed to simulate conversation with one or more human users in natural language. Providing CAs with knowledge, intelligence and humanoid interface has allowed them to be used in several practical applications. This paper presented the development and its performance of the interactive CAs called Artificial Intelligent Natural-language Identity or AINI.

Key-Words: - Conversation Agents (CAs), Artificial Intelligence (AI), Artificial Intelligent Natural-Language Identity (AINI), Natural Language Processing (NLP)

1 Introduction

During the past decade, one can observed that there are rapid advances in interactive conversation agent (CAs), spoken language technology, natural language processing and multimodal interfaces to replace CAs which based on pattern matching and typed-text-based. All these have stimulated the interest towards a more human-like conversational interfaces [1]. Many researchers have also been observed in AI researches into natural language conversation [2] and the application of CAs in interactive games [3]. They have proposed different techniques and produced several natural language conversation systems. Every year they present their work by competing for the Turing Test (TT)[4].

There is hundreds of different CAs developed for a variety of reasons. They range from hardwired programs with simple coded patterns to systems built upon embedded learning algorithms which continuously expand their language knowledge base. Our interactive CA called Artificial Intelligence Natural-language Identity (AINI) was created purely for mimic human interaction or as part of interactive games [3]. Many CAs are designed to provide specific information and direct the dialogue to specific topics.

On the other hand, conversational characters represent the convergence of animated interface agents and human-computer dialogue systems. As animated agents get more realistic, the user

naturally expects to be able to interact with them in natural language. And as human-computer dialogue systems develop, it appears that users could interact more readily in natural language if the system is personified through an agent.

2 Tricks in Interactive Conversation Agents

Some people interpret the TT as a setting in which you can "cheat". The imitation game (IG) has no rules constraining the design of the machines. Turing describes how machines could be "rigged" to overcome certain obstacles proposed by opponents of the idea that machines can think. An obvious example is about machines making mistakes. When the machine is faced with an arithmetical challenge, in order not to give away its identity of being fast and accurate, it can pause for about 30 seconds before responding and occasionally give a wrong answer. Being able to carry out arithmetical calculations fast and accurately is generally considered intelligent behaviour. However, Turing wishes to sacrifice this at the expense of human-ness. Some commentators think this is "cheating". The machine is resorting to certain "tricks" in its operations rather than imitating the human ways. However, arithmetic is a highly specific domain. Modifying the programs in this manner cannot hurt: If a machine can pass the

test, it can then be re-programmed not to cheat at arithmetic. If it does not resort to this, the interrogator can ask a difficult arithmetical problem as his/her first question and decide that he/she is dealing with a machine right then and there. We believe the best way to handle this issue is considering this as "deception" rather than as "cheating". After all, in a way, the game is all about deception. It can be seen that Turing considers it possible that a sufficiently human-like machine (i.e., a machine that is sufficiently good at playing the IG is bound to make such mistakes as we attribute to humans.

Table 1: Tricks in Conversation Agent

ELIZA	<ul style="list-style-type: none"> - Fostered by including substrings of the user's input in the program's output. User: You hate me. ELIZA: Does it please you to believe that I hate you? - Use of the Rogerian mode, which provides unimpeachable cover for the computer.
PARRY	<ul style="list-style-type: none"> - Admitting ignorance. "I don't know." - changing the level of the conversation "Why do you ask that?," - Introducing new topic, launching into a new story also called as simulates paranoid behavior.
Other Tricks	<ul style="list-style-type: none"> - Having many fragments of directed conversation stored in activation network. - Changing the level of the conversation - Humorous statements to make the program seem more human. - Agreeing with the user, in certain cases, the program can safely agree with the user rather than being non-committal. - Simulated typing, by including realistic delays between characters and imitate the rhythm of a person typing. - Make longer replies and its seemed more human-like that cur reply. - Introduce a new subject with a certain probability. - Fragments of directed conversation stored in activation network, "opening book." - Controversial statements, "People don't own cats..." - Agreeing with the user, rather than being non-committal. - Excerpting News, weather forecast, textual games, etc.

The CAs has already fools "average" questioners. If a larger collection of "tricks" sufficed, would you redefine "artificial intelligence," "average questioner," or "trick?" Perhaps the biggest obstacle to the advancing of this area is there are not many uses for fooling people besides the Turing test [5]. Fifteen years after Turing proposed the imitation game, ELIZA written by Joseph Weizenbaum [6] from MIT and PARRY[7] program from Stanford University demonstrated that "a simple computer program" could successfully play the imitation game by

resorting to a few "tricks," the most obvious being to answer questions with questions[2]. The others CAs such as PARRY, TINYMUD and Hex also have their own tricks as depicted in the Table 1.

The fact that a program with a limited handcrafted restricted knowledge bases, simple pattern recognition techniques and a few linguistic or AI tricks will be unable to hold a long term conversation with a human. Therefore, this paper addresses the above challenge and introduces a novel design of the interactive CAs aiming to overcome these restrictions.

3 Conversation Agent Architecture

In the design of CAs such as TARA (Terrorism Activity Resource Application) [8], CAs were experimented with using a two-tiered architecture based on the original ALICE's Loebner Prize winning Program D [9] on Java. This was followed by CMU Nursebot called Pearl [10, 11] and OpenMindBot [12], CatBot [13], TutorBot [14], Partner (SP) [15, 16], ChatBot [17] and Persona-AIML [18].

While many techniques and programming languages have been proposed over the years to develop CAs, the primary challenge remains in how to overcome the poor scalability and the lack of flexibility to handle the heterogeneous CAs software designs. Therefore, the proposed approach is to employ a novel *N*-tiered architecture similar to the *service-oriented application architecture* to capture information at decreasing levels of granularity at the application server tier, client tier and agent data server tier. This architecture enables us to handle dynamic features unique to scripting languages, such as dynamic typing and code inclusion, which have not been adequately addressed by previous techniques.

3.1 AINI's *N*-tiered Architecture

The key to success in future conversation systems is to develop general dialogue systems [19]. This can be defined as a framework that is not designed for a particular application, but can be 'plugged' to various applications. It should require minimal effort to develop the domain knowledge and to adapt to different applications. These considerations have led to the proposal of two main tiers for the handling of the generality issue in the CA framework. They are the *Channel Service Tier* and *Domain Service Tier* as shown in Fig. 1.

These additional tiers contain all the necessary logic to receive and delegate the requests. In the

previous three-tiered architectural concept, it was illustrated that the architecture is lacking in two major aspects. First, the three-tiered architecture combines communication logic with business logic in the application server tier to handle all the requests by using a single interface to access services. Second, in the three-tiered architecture, clients and developers are faced with potential issues with the communication channels. In order to avoid these consequences, the *Channel Service Tier* has been used, as it allows the creation of a flexible and extensible architecture able to support different interfaces and additional services more easily. The *Domain Service Tier* will support a flexible and extensible architecture that is capable of providing any domain application based on a client's request, without the necessity of changing domain services.

3.1.1 Channel Service Tier

Building extensible and scalable systems is a challenging task. Currently, different clients communicating over different channels or protocols such as HTTP or TCP, must be offered the same set of application services by the system. In addition, the current deficiency of CAs architecture is the lack of universally agreed standards. In general, the number of channels and the number of demanded services are expected to increase over time, and a gradual shift from web-based to mobile-based applications. The proposed AINI's *N*-tiered architecture could be a viable solution to meet these

future challenges. Each channel has its own communication protocol. As the application tier has to satisfy requests from clients such as web browsers, mobile browsers, MSN Messenger interface applications, it must deal with service requests coming from different channels and each channel is characterised by its own communication protocol. Therefore, a server residing on the application tier must be able to support these communication protocols.

3.1.2 Domain Service Tier

Under the standard *N*-tiered architecture, the application tier and the data access tier are flexible and scalable toward domain knowledge independency. The suggestion made here is to create an additional layer called the "domain service tier" to resolve the coupling issues. The role of the domain service tier will be focused on handling concurrent access and managing changes to the domain knowledge tier. This domain service tier prepares a number of its own services that override methods or features of the database using SQL query. It is also the responsibility of the domain service tier to choose either the domain-specific or open-domain knowledge. This solution increases the cohesion of each tier and prevents the overall system from becoming excessively complex. In addition, it will assist the development of the domain application. For instance, to deploy a furniture domain application such as the Anna CA

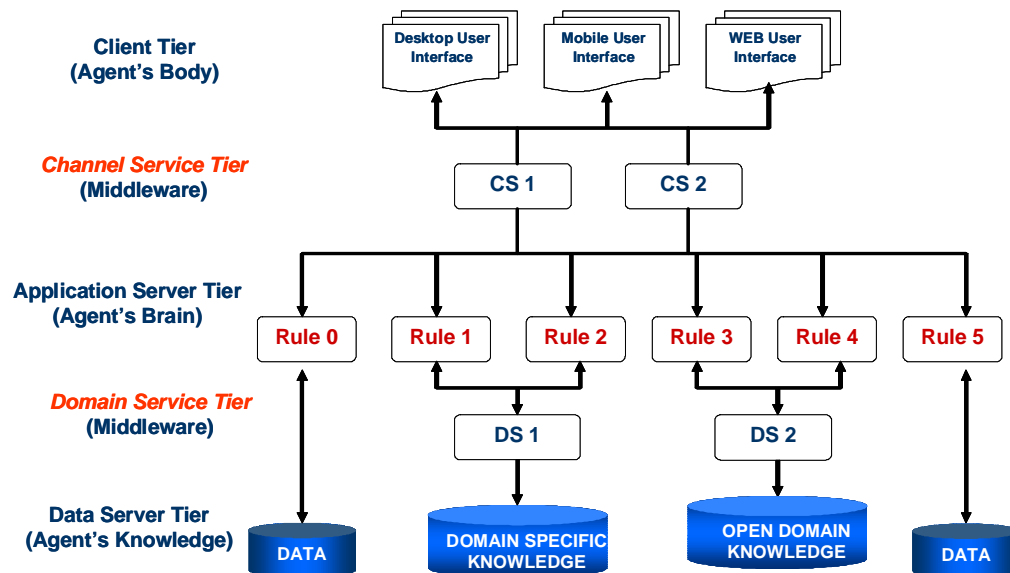


Fig. 1: *N*-tiered Architecture with addition Channel Service Tier and Domain Service Tier

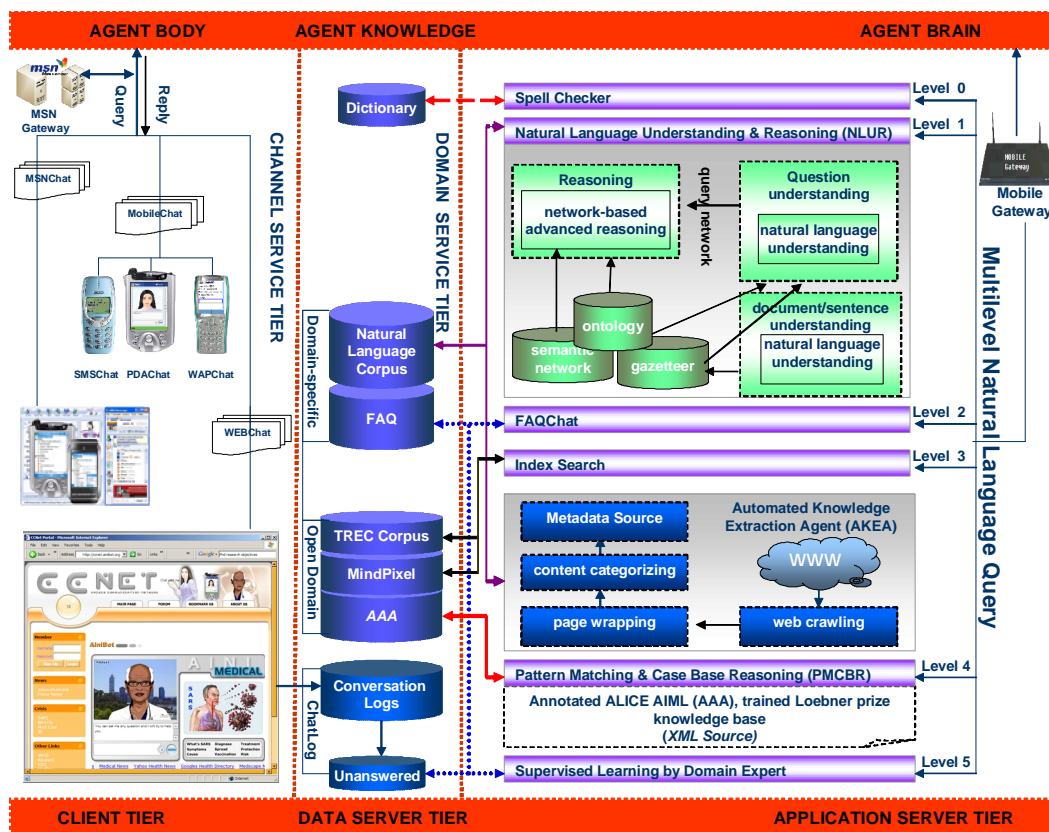


Fig. 2: AINI's Conversation Agent Architecture

used by IKEA, only domain-specific need to be updated, instead of the entire data server tier.

The AINI conversation architecture has been reported in previous publications [20, 21]. As illustrated in Fig. 2, AINI employs a *N*-tiered architecture that can be configured to work with any web, mobile or other network applications. It comprises a client tier (hereafter called “*agent body*”), an application server tier (hereafter called “*agent brain*”) and a data server tier (hereafter called “*agent knowledge*”).

The *channel service tier* and the *domain service tier* are designed to support different channels of interfaces and can quickly be augmented with domain knowledge for specific purposes. The architecture provides features of multimodal interface, multilevel natural language query and multiple knowledge bases. The process of communication and answering is as follows. Given a question, AINI's agent body will perform a HTTP over TCP request from a Web, mobile service or instant messaging service to the agent brain. Here, the agent brain will attempt to formulate a reply to the sentence via the Natural Language Understanding and Reasoning Module. This module will do the sentence parsing to

produce grammatical categories and grammatical relationships by extracting pertinent information through the agent's knowledge, such as Noun Phrase and Verb Phrase. If this step is successfully parsed, the network-to-path reduction will be carried out by Network-based advanced reasoning. From the query network, the *question* is decomposed into sequences of words or phrases and they form the nodes of a tree. After obtaining the query network, the task of answering the question is reduced to discovering the presence of the query network in the whole semantic network. Then, the problem of discovering the *answer* has been reduced. The appropriateness of answer is achieved through the logic and network approach during answer discovery using a template-based approach. The template method matches the answer against question templates requested by the user. The system will then produce an answer in the agent body. Each of the agent tasks is described in further detail in the following section.

3.2 Agent Body (Client Tier)

The user interface, or human-computer interface (HCI) [22], resides in the agent body and it supports three different types of channels of

communication, such as Webchat, MobileChat or MSNChat, controlled by the *channel service tier*. AINI uses HTTP over TCP to connect to the Internet and mobile services to communicate with the users. The user interface was written using HTML, Javascript, vbscript, XML, WML, Flash Action Script and other client side scripting languages.

WebChat is web-based interface that allows users to interact in real-time with AINI through cross-browser including Internet Explorer, Mozilla Firefox, Safari, Opera, and Netscape. For a few years, the CAs emphasized the ‘embodiment’ feature of the interactive virtual characters or web robots significant progress was made in terms of software architectures [23]. To foster a relationship with the user and to encourage the user to interact with the system in a natural manner, it is possible to incorporate an anthropomorphic agent or avatar in the AINI’s framework. Nass and Reeves [24] from Stanford University suggested strongly that there is a tendency to treat computers as a human entity, which in turn results in human based social responses. Cassell et al. [25] pointed out that conversational computer agents have played very important roles in human-computer interactions. Such skills include the abilities to use face, hands and tone of voice to regulate the process of conversation, as well as the ability to use verbal and non-verbal means. To achieve this end, Microsoft Agent Technology, Flash Technology and Oddcast Technology¹ are used. These technologies introduce the use of an animated and speech enabled avatar, which will serve as the medium between the system and the user. Besides, users also can go through all the information on the website for the topics they are interested in. At the same time, they can place questions to AINI for more information or guidance. Another advantage of the WebChat is a collaborative browser which allows a portal to guide the users through the website of the organization by automatically “pushing” URLs and information from other websites to the user’s browser. This not only facilitates communication between CAs and users, but also allows the intelligent CA to help users

locate specific information on their websites. In addition, users are also able to personalize their WebChat interface by customizing the avatars, in such ways as choosing their avatar’s gender, voice, face, clothing or accessories as shown in Fig. 3 which is powered by Flash Technology² or Microsoft Agent Technology³ as shown in Fig. 4. AINI’s architecture is scalable and can be embedded into any existing website without changing its code.



Choose your favourite avatar, gender, hair style, clothes, classes, etc.

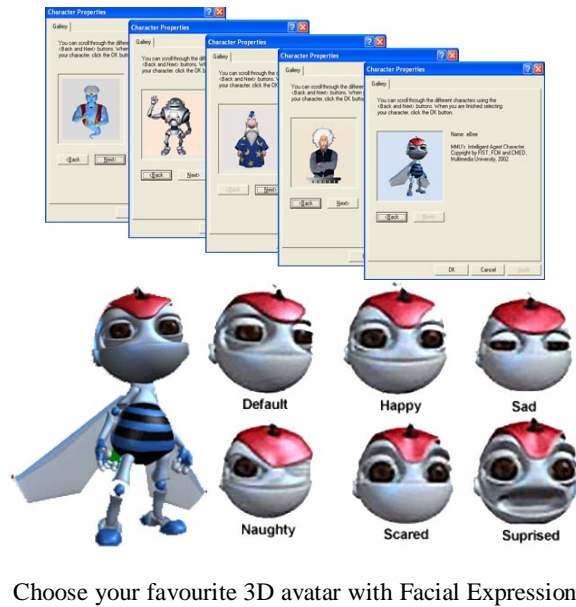


Fig. 3: An illustration on how AINI could be embedded in a Crisis Communication Website with personalized avatar powered by Flash Technology

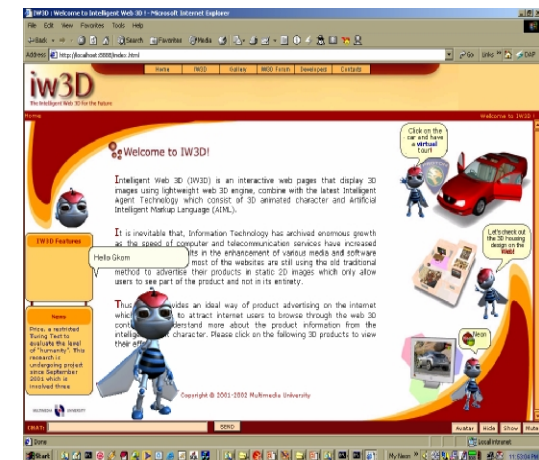
¹ Crisis communication research on SARS was supported by Oddcast Inc. in 2003 and Bird Flu in 2005. In these projects, Oddcast provide a streaming voice through text-to-speech (TTS) solution to make our interactive CAs more presentable. Oddcast interface that allows us to create and embed customize animated characters within AINI’s N-tiered architecture.

² <http://www.adobe.com/products/flash/>

³ <http://www.microsoft.com/msagent/>



Choose your favourite 3D avatar with Facial Expression



to having a more human and personalised interface between a computer and human. The PDACHat with AINI performs functions similar to web chats, but in a mobile environment. It is a prototype designed to blend mobile technology with natural language to help humans interact more naturally with mobile devices [26]. An example of PDACHat is shown in Fig. 5. The PDACHat was designed using WiFi technology and powered by Microsoft Windows Mobile Technology⁴ embedded with Pocket Internet Explorer on a HP iPAQ Pocket PC. The computer-generated avatar of the CAs is displayed on the PDA using Adobe Flash for Pocket PC⁵, and the system uses a WiFi wireless connection to AINI's server.



Fig. 5: An illustration on how AINI could be embedded in PDA with personalized avatar power by Microsoft Windows Mobile Technology

Fig. 4: An illustration on how AINI could be embedded in a Intelligent Web 3D Website with personalized 3D avatar powered by Microsoft Technology

Conversation chat through messaging applications were the first and most successful community applications for mobile services such as WAP, GPRS and 3G extended by Web services. In AINI's framework, users can freely select the CA they prefer to access. Hence, they can chat anywhere, at any time, with any device. A mobile chatting module is implemented in a series of logical phases. Today, around one trillion text messages are sent each year, and this number is growing. Here, AINI's N-tiered architecture is ready to support this technology by integrated MobileChat module on the PDA or mobile phone.

The idea of developing AINI into Personal Digital Assistance (PDA) is an interesting approach

3.3 Agent Knowledge (Data Server Tier)

Another significant difference between this research and other research on CAs is the domain knowledge model. Dahlbäck and Jönsson [27] stressed that the domain model represents the structure of the knowledge which comprises a subset of general knowledge. Such systems normally are comprised with two subcategories: the *traditional/narrow domain* or *domain-specific*, and the *open-domain*. In the traditional domain, systems attempt conversational fluency based on limited domains of expertise. ELIZA [28], for example, simulates a Rogerian psychotherapist, and its implementation is commonly known as DOCTOR and PARRY[7]. DOCTOR and

⁴ <http://www.microsoft.com/windowsmobile>

⁵ <http://www.adobe.com/software/flashplayer/pocketpc>

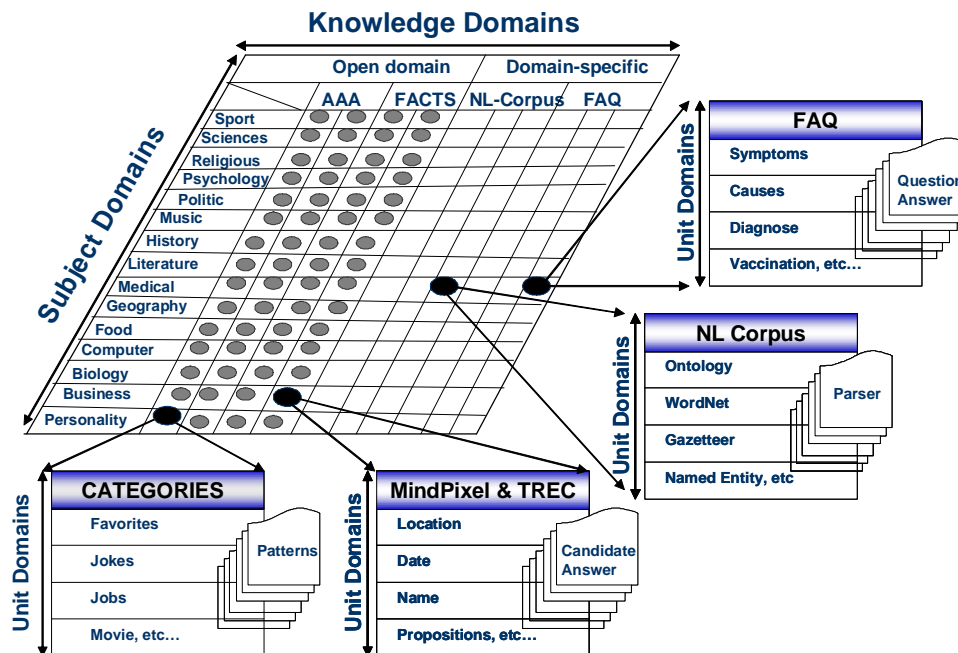


Fig. 6: Domain Knowledge Matrix Model (DKMM)

PARRY's domain was restricted to paranoid hospital patient expressions. SHRDLU [29] is another program simulating a CA which is able to interact within a simple world knowledge of "blocks". SHRDLU was an entry in an early Loebner Prize competition, where the evaluation was based on the restricted tasks [30]. However, in the Fifth Annual Loebner Prize Contest in 1995, the Loebner prize criteria were changed to include unrestricted domains [31], requiring computer entries to converse indefinitely with no topic restrictions.

Hence, it is understood that general purpose CAs are not necessarily able to answer questions on a specific domain subject. On the other hand, domain-specific systems lack the flexibility to handle common sense questions. To overcome the above limitations, we proposed the Domain Knowledge Matrix Model (DKMM) [32]. The data server layer serves as storage for data and knowledge required by the system. This is where AINI's conversational knowledge bases are stored. It is well understood that true intelligent action requires large quantities of knowledge. Such a reservoir of knowledge can be harvested from the internet and deployed in the domain matrix knowledge bases' architecture. This forms the basis for the construction of large-scale knowledge bases to be used as the engine for intelligent conversation systems. AINI is the mechanism used to manage

the knowledge and to provide appropriate answers to the user.

AINI's DKMM incorporates several knowledge subjects. This is analogous to the consultation of expertise knowledge from multiple experts. For example, a *sales* knowledge domain should contain expertise on how to improve sales. However a sales person is expected to have a wide range of common sense which enable CAs have ability to engage the potential customer in general conversation. Hence, an intelligent system should also incorporate open-domain knowledge to handle general or generic questions. By including multiple domain knowledge bases within AINI's single knowledge domain, the proposed AINI will be able to hold "meaningful" and prolonged the conversations with the users.

In this proposed DKMM [32], both the open-domain and domain-specific knowledge bases are predefined in the agent's knowledge. These modules are used to support the various knowledge levels at the agent brain tier. Depending on the user's input, the agent will respond or switch from one level to another in the agent brain. While the system is capable to communicate with the user beyond the knowledge domain, there are cases where the system will exhaust its capability to answer the queries. In such case, the system will attempt to divert the focus back to the current topic of interest by responding with some predefined

random statements. The purpose is to direct the user's attention back to the system's domain-specific state. Hence, AINI will attempt to "cycle" between the six levels of information processing within the agent brain tier (discussed in Section 3.4) supported by the various knowledge modules in the agent knowledge tier.

A way to view the proposed DKMM is given in Fig. 6. In this approach, the knowledge base of the AINI can be considered as a collection of specific conversation domain units. Each unit handles a specific body of knowledge used during the conversation between AINI and the user. The knowledge can be seen as arranged in the vertical columns making up the open-domain or domain-specific knowledge. In addition, specific subjects are shown in the horizontal rows. For example, in the open-domain knowledge, the subject units will cover topics such as personality, business, biology, computers, etc. In this research, our focus is on the subject of medicine; and in particular, the bird flu pandemic. Therefore, additional bird flu domain knowledge is being incorporated in the domain-specific row "medical", and column NL-Corpus.

In this research, the novel contribution is the development of the DKMM, which is a "domain knowledge plug-in module" through the domain service tier. With this arrangement, the domain-specific knowledge and open-domain knowledge could become portable, scalable and incorporated easily with other domain applications. This approach will also allow future improvements to encourage collaborative contribution to the other domain applications and tasks.

Currently⁶ AINI's open-domain knowledge base has more than 160,000 entries in the common sense stimulus-response categories. Of these, 100,000 came from MindPixel, 997 factoid questions from the TREC training corpus and 45,318 categories from the AAA knowledge bases. On the domain-specific knowledge base, AINI has more than 1,000 online documents extracted by AKEA. This makes up over 160,000 stimulus-response items in total. AINI also has 158 FAQ pairs of questions and answers, which have been updated using AKEA. In addition, AINI has also collected more than 52,890 utterances in conversations with online users since the first prototype of AINI was put online in the February 2006[32]. These utterances will be integrated into AINI's knowledge bases through supervised

learning by domain experts. At present, AINI has learnt about 5,000 categories from conversations with online users. All of this combined knowledge has made up the total of 161,473 stimulus response categories in AINI's knowledge bases. To compare AINI with other systems, the original conversation programs such as ELIZA, written by Professor Joseph Weizenbaum of MIT, has only 200 stimulus response categories. ALICE Silver Edition was ranked the "most human" computer, and has about 120,000 categories, which include 80,000 taken from MindPixel as summarised in Table 2.

Table 2: AINI's Stimulus-response Categories

Domain Knowledge	Sources	Categories	%
Domain-Specific	NL Corpus	10,000	6.19
	FAQ	158	0.10
Open-Domain	MindPixel	100,000	61.93
	TREC Corpus	997	0.62
	AAA	45,318	28.07
Supervised Learning	Conversation Logs	5,000	3.10
TOTAL		161,473	100

3.4 Agent Brain (Application Server Tier)

In the classical CA design, the original ELIZA's brain used MAD-Slip programming language. The technique that is in use in a "CAs database" or "script file" to represent the CA knowledge is known as Case-Based Reasoning (CBR) [28]. The original ELIZA had only about 200 rules, and 90% of ELIZA's stimulus-responses are found in the associated script file. However, these limitations have been surpassed by Loebner Prize [4] winning CA, which are smarter than before. More new features have been added since the last submission. The "script file" which acts as an agent brain for the CAs has been completely rewritten, and it is definitely better than in the previous versions of the program. This includes ALICE, one of the 'most-human' natural language CAs, which uses a programming language called AIML that is specific to its program, and its various clones, named ALICEBots [33]. Nevertheless, ALICE is still based on pattern matching and case-based reasoning (PMCBR), whereas natural language understanding and reasoning is not available in ALICE. This is the same technique that ELIZA, the first CA, was using back in 1966.

In AINI, the communication with users takes place through natural language query. AINI's brain

⁶ Till 1 August 2007, AINI's have 161,473 stimulus-response categories in their knowledge base.

implements its decision making network based on the information it encounters in the six levels of natural language modules, as have been shown in Fig. 7 and discussed in references [34]. The input and output of each module is an XML-encoded data structure that keeps track of the current computational state. The knowledge modules can be considered as transformations over this XML data structure. The system accepts queries from the users and it processes the queries based on the information contained in AINI's knowledge bases.

The agent brain tier handles the process of the queries or business logic. Here, one or more domain service tiers are configured to compute the dialogue logic through the multilevel natural language query algorithm. In this tier, it is based on a goal-driven or top-down natural language query (NL-Query) approach, which is similar to the way that humans process their language. As indicated by literature in the field of Natural Language Processing (NLP), the top-down approach is by far the best approach. Mentalese, or 'language of thought', and conceptual representation support the ideas of a top-down approach [35]. This was also supported by research in natural language understanding [36], rhetorical structure theory [37], summarisation [38], plan-based approaches [39], and SHRDLU [29] the first CA to use NLU, are examples of top-down approaches. Therefore, AINI's agent brain uses a top-down NL-query approach to simulate human conversation. However, in the robotic design, the MIT Cog Robot

research fervently supports the bottom-up approach when modelling the human brain.

As shown in Fig. 7, the top-down multilevel natural language query approach consists of six levels of queries, namely Spell Checker (Level 0), Natural Language Understanding and Reasoning (NLUR) (Level 1), FAQChat (Level 2), Index Search (Level 3), Pattern Matching and Case-based Reasoning (PMCBR) (Level 4) and Supervised Learning (Level 5). All these levels were designed modularly and can be plugged in easily without the changing of the entire codes.

In summary, the multilevel natural language query approach includes the following stages:

- **Level 0** - AINI parses the user's input by reading in sentences one by one, and then each sentence is converted to a list of words to be processed by the spell checker. The spell checker will propose alternative words from the dictionary if it finds any typo errors. If this step is passed, AINI would attempt to formulate a reply to the sentence by calling the next module;
- **Level 1** - The sentence could be sent to the NLUR module. This module will do the sentence parsing to produce grammatical categories and grammatical relationships such as Noun Phrase and Verb Phrase. Name-Entity Recognition will then identify each of the keywords to look for a match in the gazetteer

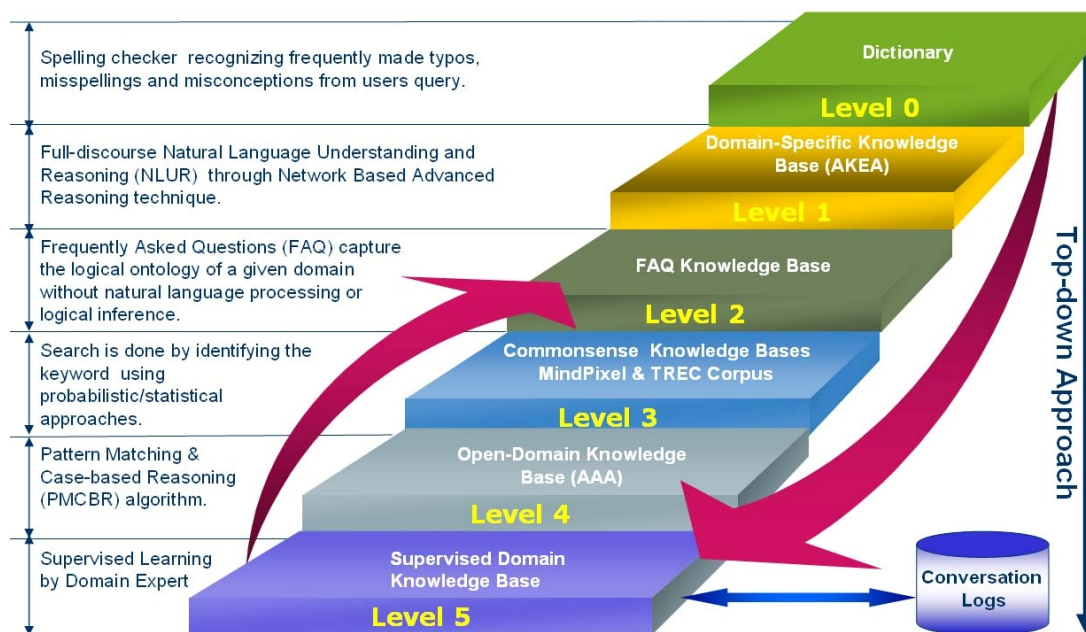


Fig. 7: Multilevel Natural Language Query

for their relation inference and discourse integration. If this step is successfully parsed, the network-to-path reduction will be carried out by network-based advanced reasoning. From the query network, the set of sequences *question* from the leaf node to the root node is obtained.

- **Level 2** - If an empty string is returned from the database, AINI will then go to the FAQ Module using an FAQ question-answering system. In this stage, sophisticated natural language processing or logical inference is ignored as they have already been performed in the previous module.
- **Level 3** - If no database match was found, AINI will then go to the next module, called Index Search. This module relies on the application of a mix of linguistic rules and probabilistic or statistical principles.
- **Level 4** - If this also fails, AINI will then try to detect whether the sentence was a trick question. It would look for the PMCBR module, which is commonly found in ELIZA and ALICE. In this case, AIML technology will respond with suitable witty replies.
- **Level 5** - If even this module fails to work (which could happen if all of its witticisms had already been used), then AINI's routine would be invoked to randomly generate a reply. In this stage, the result checking will still be the responsibility of the domain expert, who will be able to correct errors and to pick a proper matching response from a list of possible matches from the conversation log which is unanswered by AINI. This undoubtedly will lead to better performance and accuracy in future conversations.

4 An Example of the Proxy Conversation

In our experiment⁷, we created an amusing and informative example of how CAs converse between them in the wild with the simulated "conversation" between Eliza⁸, ALICE⁹ and AINI as depicted in the Fig. 8.

```

Eliza :Hello. I am ELIZA. How can I help
       you?
AINI  :Hi there! Hi, ELIZA.
ALICE :OK I will call you ELIZA. Just keep
       talking to me as long as you can.
AINI  :My name is Aini. Gosh! Chat amongst
       yourselves.
ALICE :I'm pleased to introduce myself to you,
       Aini. Do you have any conditions I
       should know about?
Eliza :Oh, i have any conditions you should
       know about.
AINI  :Aha. Cool. I have a friend named
       E.L.V.I.S.
ALICE :Next question? That's nice. I have a
       very nice programmer.
AINI  :Do you have any hobbies? You can click
       on the button on the bottom right corner
       to get more news. I have a great
       botmaster.
Eliza : Oh, i can click on the button on the
       bottom right corner to get more news you
       have a great botmaster.
ALICE : And? Oh you can, can you?
Eliza :Oh, i can can i.
AINI  :Oh you can, can you?
ALICE :I see. No I don't think I can do it.

ELIZA : Why do you not think you can do it?

```

Fig. 8: Conversation logs between three Conversation Agents

In this conversation, most of the AINI's knowledge is extracted from the general knowledge base. This is because, during the conversation between CAs, majority of the questions are based on the open-domain knowledge base, instead of domain-specific. In addition, from the conversation logs, how well does the CAs really remember the thread of your conversation? Many CAs will ask your name, and incorporate this into future responses. Some CAs also attempts to identify the "topic" of conversation, and a few will be able to tell you what the topic is when you ask about it (although even this is rare). Just remember that asking your name or any other piece of information and storing it in a variable for later use doesn't constitute "understanding" what's going on in any sophisticated sense, nor is it the basis for "reasoning" in any sense of the word. Many companies try to pass off tricks like this as a kind of "memory" or "knowledge of context".

In the last conversation, you could see how well does the CAs really understand "anaphora"? Anaphora is references to previously-named things. In English, pronouns like "I", "me" "you", "he", "she", "it", and so on usually refer to previously

⁷ The experiments conducted in this paper were performed by the authors in the period of January – June 2007.

⁸ <http://www-ai.ijs.si/ELIZA/ELIZA.html>

⁹ <http://www.alicebot.org>

named people or things. Some CAs companies will pretend that their software can really identify the target of an anaphoric reference. But such claims can be proven false, and can easily be shown to be keyword-based as well. You need only pursue a dialog a few turns to show that a CAs really has no clue what you're talking about, beyond a rudimentary index of the "current topic". You can try all the tricks you want; some of them will get you witty answers, a few will even be answered "correctly" (until you twist the trick a bit more), and most will just get you a plain stupid answer.

5. Conclusion

This paper has given an overview the design of the interactive conversation agents. This paper also provided an analysis and development of the interactive CAs and shows the feasibility of our own CAs called AINI. From the discussion, it was found that multilevel NL-Query models, DKMM, and *N*-tiered architecture address the extensive variability that is encountered in today's CA frameworks. There is a delicate balance between flexibility and simplicity. Performance cannot be compromised for the sake of flexibility and the lowest common denominator solution is often unacceptable. It is necessary to have flexible development environments, tools, and regression tests. Application interoperability is an essential direction for future improvement of the interactive CAs design.

References:

- [1] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, *Embodied Conversational Agents*. Cambridge, MA: MIT Press, 2000.
- [2] J. Weizenbaum, "ELIZA - A computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, pp. 36-45, 1966.
- [3] O. S. Goh and C. C. Fung, "AINI - Embodied Conversation Agent Applicable for Interactive Games," in *7th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS '08)*, Hangzhou China, 2008.
- [4] H. G. Loebner, "Home page of the Loebner prize—the first Turing test," Loebner Prize, 25 July 2007, <http://www.loebner.net/Prize/loebner-prize.html>.
- [5] M. L. Mauldin, "Chatterbots, Tinymuds, And The Turing Test: Entering The Loebner Prize Competition," in *AAAI-94*, 1994.
- [6] J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, pp. 36-45, 1966.
- [7] K. M. Colby, *Artificial Paranoia: A Computer Simulation of Paranoid Processes*: Pergamon Press Inc., 1975.
- [8] R. Schumaker, Y. Liu, M. Ginsburg, and H. Chen, "Evaluating the Efficacy of a Terrorism Question Answer System: The TARA Project," *Communications of the ACM*, vol. 50, pp. 74-80, 2007.
- [9] N. Bush, "Getting Started With Program D," ALICE Foundation, 30 July 2007, <http://www.alicebot.org/resources/programd/readme.html>.
- [10] C. Torrey, A. Powers, S. Fussell, and S. Kiesler, "Exploring Adaptive Dialogue Based on a Robot's Awareness of Human Gaze and Task Progress," in *Human-Robot Interaction (HRI'07)*, 2007.
- [11] A. Powers, A. D. Kramer, S. Lim, J. Kuo, and S. Sau-lai LeeKiesler, "Eliciting information from people with a gendered humanoid robot," in *IEEE International Workshop on Robot and Human Interactive Communication*, Roman, 2005, pp. 158 - 163.
- [12] S. Rao, S. Garg, H. Lieberman, and H. Liu, "Commonsense via Instant Messaging," Massachusetts Institute of Technology, 20 July 2007, http://ocw.mit.edu/NR/temp/5C7CF160-84C0-4E56-BD32-5F36B6CCE2AE/0/proj_file9_wor.pdf.
- [13] K. Brennan, "The Managed Teacher: Emotional labour, Education and Technology," *Educational Insights*, vol. 10, 2006.
- [14] O. D. Pietro, M. D. Rose, and G. Frontera, "Automatic Update of AIML Knowledge Base in E-Learning Environment," in *Computers and Advanced Technology in Education*, 2005.
- [15] F. Meijerink, P. Maanen, P. v. Maanen, A. J. v. Vliet, and A. Nijholt, "Disclosure with an Emotional Intelligent Synthetic Partner ,," in *Workshop: Tools for Psychological Support during Exploration Missions to Mars and Moon, European Space Research and Technology Centre (ESTEC)*, Noordwijk, The Netherlands, 2007.
- [16] S. Kshirsagar and N. Magnenat-Thalmann, "Virtual Humans Personified," in *Autonomous*

- Agents Conference (AAMAS) 2002*, 2002, pp. 356-359.
- [17]C.-H. Lu, G.-F. Chiou, M.-Y. Day, and C.-S. Ong, "Using Instant Messaging to Provide an Intelligent Learning Environment," in *ITS 2006*, Berlin Heidelberg, 2006, pp. 575 – 583.
- [18]A. M. Galvao, F. A. Barros, A. M. M. Neves, and G. L. Ramalho, "Persona-AIML: an architecture for developing chatterbots with personality," in *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, 2004, pp. 1266 - 1267.
- [19]J. F. Allen, D. K. Byron, M. O. Dzikosvska, G. Fergusson, L. Galescu, and A. Stent, "Towards conversational Human-Computer Interaction," in *AI magazine*, 2001.
- [20]O. S. Goh, C. C. Fung, A. Depickere, and K. W. Wong, "An Analysis of Man-Machine Interaction in Instant Messenger," in *Advances in Communication Systems and Electrical Engineering*. vol. 4, X. Huang, Y.-S. Chen, and S.-I. Ao, Eds. US: Springer, 2008, pp. 197-210.
- [21]O. S. Goh, C. C. Fung, and K. W. Wong, "Query Based Intelligent Web Interaction with Real World Knowledge," *New Generation Computing*, vol. 26, pp. 3-22, 2008.
- [22]C. Ilioudi, A. Lazakidou, and A. Andreou, "Study of Social, Ethical and Human-Computer Interaction Aspects of the MEDARIST II System," *WSEAS Transactions on Computers*, vol. 6, pp. 2108-2112, 2004.
- [23]P. Cosi, C. Drioli, F. Tesser, and G. Tisato, "INTERFACE toolkit: a new tool for building IVAs," in *Intelligent Virtual Agents Conference (IVA'05)*, KOS Greece, 2005.
- [24]C. S. Nass, J. Tauber, and R. Ellen, "Computers are Social Actors," in *CHI'94: Human Factors in Computing Systems*, Boston, MA, 1994, pp. 72-77.
- [25]J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, and H. Vilhjalmsson, "Embodiment in conversational interfaces: Rea," in *the CHI 99 Conference on Human Factors in Computing Systems*, New York, 1999.
- [26]G. Mavrommatis, A. Lazakidou, K. Siassiakos, and G. Lazakidou-Kafetzi, "An Educational Help Desk Model for Mobile Environments," *WSEAS Transactions on Advances in Engineering Education Journal*, vol. 6, pp. 469-474, 2006.
- [27]N. Dahlbäck and A. Jönsson, "Integrating domain specific focusing in dialogue models," in *Eurospeech'97*, Rhodes, Greece, 1997, pp.,2215-2218.
- [28]J. Weizenbaum, "ELIZA—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, pp. 36-45, 1966.
- [29]T. Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," *Cognitive Psychology*, vol. 3, 1972.
- [30]S. M. Shieber, "Lessons from a Restricted Turing Test," *Communications of the ACM*, vol. 37, pp. 70-78, 1994.
- [31]H. G. Loebner, "In Response [to Shieber 1994a]," *Communications of the ACM*, vol. 37, pp. 79–82, 1994.
- [32]O. S. Goh, C. C. Fung, and A. Depickere, "Domain Knowledge Query Conversational Bot in Instant Messaging (IM)," *Knowledge-Based Systems*, vol. 21, 2008.
- [33]R. S. Wallace, H. Tomabechi, and D. Aimless, "Chatterbots Go Native: Considerations for an eco-system fostering the development of artificial life forms in a human world," *Pandorabots*, 20 July 2007, http://www.pandorabots.com/pandora/pics/chat_terbotsgonative.doc.
- [34]O. S. Goh, C. C. Fung, K. W. Wong, and A. Depickere, "Multilevel Natural Language Query Approach for Conversational Agent System," *International Journal of Computer Science*, vol. 33, pp. 7-13, 2007.
- [35]J. A. Fodor, *Elm and the Expert: An Introduction to Mentalese and Its Semantics*: Cambridge University Press, 1994.
- [36]M. Stanojevic and S. Vranes, "Applying Connectionist Model to Natural Language Understanding," *WSEAS Transaction on Information Science and Applications*, vol. 8, pp. 1501 - 1507, 2006.
- [37]J. D. Moore and C. L. Paris, "Planning text for advisory dialogues: Capturing intentional and rhetorical information," *Computational Linguistics*, vol. 19, pp. 651-695, 1992.
- [38]D. Radev and K. McKeown, "Generating natural language summaries from multiple on-line sources," *Computational Linguistics*, vol. 24, pp. 469-500, 1998.
- [39]E. Reiter and R. Dale, *Building Natural Language Generation Systems*: Cambridge Univ. Press, 2000.