SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 4th Annual Sessions

31st October 2007

Sri Lanka Association for Artificial Intelligence – 2007

Proceedings of the 4th Annual Sessions
31st October 2007
Moratuwa
Sri Lanka

Editors:

Prof. Asoka S. Karunananda
Dr. Nalin Wickramaarchchi
Ms. Menaka Ranasingha

Sri Lanka Association for Artificial Intelligence, Department of Mathematics & Computer Science, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka.

Table of Contents

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Keynote Address

Artificial Intelligence and the Sri Lankan research community: A personal perspective

Professor H Sriyananda

I was a little surprised, but very grateful, at being asked to speak to you this morning. I was told that I could select the topic of my address and this did give rise to some difficulty. I spent a lot of time thinking about what I should say, and finally decided that I am likely to be excused for speaking on things that I know nothing, or very little about.

Artificial Intelligence is a very misleading pair of words. It can even be called an oxymoron. Artificial, in this context, means 'man-made', and intelligence, in any context, refers to a characteristic of the mind. Jean Piaget has defined intelligence as 'what you use when you don't know what to do'. There are serious and well articulated doubts expressed about the ability of a machine to possess intelligence in the normal sense of the word, for example, by Roger Penrose in his book 'The Emperor's New Mind'. Failing to understand the limits of what is possible did lead to initial optimism (in the 1950s) and subsequent disappointment about AI. Today, it has achieved a lot, especially in the way of specific applications, but there is much more reservation about the theoretical possibilities.

The Sri Lanka Association for Artificial Intelligence is, I quote 'A scientific association devoted to understanding of the mechanisms underlying thoughts and intelligent behaviour and their emulation in machines'. This of course is perhaps the most challenging area within the discipline of AI but is a possible task, for it aims at only 'emulation' and not 'embedding' human thought processes in a machine.

Areas of current interest in AI research, as extracted from the literature, are:

Social intelligence design,
Agent-based applications in complex systems,
Rough set theory and granular computing,
Chance discovery, and
Knowledge discovery and data mining.

These are not really independent areas of research in a very strict sense, and are generally overlapping and complementing each other. As an example, it would be possible to use agents in social intelligence design. There could also be other areas, depending on the classification used. They of course all relate to what are generally called complex systems. Science Vol. 284. No. 5411 (1999) listed a number of different definitions of what complex systems are:

> A complex system is a highly structured system, which shows structure with variations (Goldenfeld and Kadanoff)
>
> A complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve (Whitesides and Ismagilov)
>
> A complex system is one that by design or function or both is difficult to understand and verify (Weng, Bhalla and Iyengar)

> A complex system is one in which there are multiple interactions between many different components (D. Rind)
>
> Complex systems are systems in process that constantly evolve and unfold over time (W. Brian Arthur).

Gary Flake (*The Computational Beauty of Nature*, MIT Press, 1998) lists two essential properties of a complex system:

> The system is composed of interacting units;
>
> The system exhibits emergent properties, that is, properties arising from the interactions of the units that are not properties of the individual units themselves.

Other definitions, some of which are in conflict with at least some of the above are also found in the literature. In particular, one definition suggests that complex systems are not sensitive to initial conditions and are generally robust over long periods of time. Others suggest that they are conditioned by their overall history, rather than their 'initial conditions' and that they do evolve over time as self-organising systems. One characteristic, rather than a definition, that seems to describe complex systems is that they are not amenable to study using the traditional reductionist approach of modern science.

I have my doubts about the legitimacy and the desirability of research in social intelligence design, given the balance of power in the present (and indeed any past) society. Social intelligence design, in this context, is usually a computer mediated attempt at influencing societal values, perceptions and preferences. It of course has very legitimate applications, say, in mass education, especially distance education.
However, it is likely that any new developments in this area would be very quickly appropriated by the economically and politically powerful for their own advancement, to the detriment of the people as a whole. This is the very nature of the market economy on which the world is now modelled. Already, powerful industries are known to manipulate the market and influence people through various means, some ostensibly legal and some not so legal. SID is, even when it is not so called, a powerful tool in the hands of the advertising fraternity in influencing public opinion. The pharmaceutical industry is notorious for persuading the public to buy expensive and useless, sometimes even positively harmful, medical preparations for its own profit. (Please read Prof. Carlo Fonseka's Senaka Bibile memorial address delivered on 29th September 2007.) Similar actions by the Tobacco industry and the Baby Food industry are well documented.

I am reminded of two works of fiction which bring out this aspect very vividly, The first is Bertolt Brecht's 'Life of Galileo'. Galileo is generally considered to be the father of Modern Science, but Brecht presents the thesis that Galileo, through his public recantation to be allowed to continue with his work, did actually committed the 'original sin' of science and initiated the process by which the results of science are allowed to be appropriated by the powerful. The other is Aldous Huxley's 'Brave new world' where human beings are bred into identified social and work categories, to be happy in their station. This of course is the ultimate in the misuse of science and technology, much more than the atomic bomb.

This is really an issue of ethics and of the conscience. The Pugwash movement was born out of a similar dilemma faced by a group of scientists, mainly physicists, at the end of the Second World War, centered around the use and spread of nuclear weapons. Dangerous as the use of nuclear weapons are, the destabilizing effect and consequent tendency to violence caused by the concentration of power in the hands of a few corporations is equally dangerous to humanity. Scientists working in sensitive areas, which now

include pharmaceuticals, information and knowledge transactions, should ensure that the products of their labour are not misused.

I would like to briefly examine some of the other areas mentioned at the beginning, for a different purpose. The 'different purpose' is to suggest that we draw inspiration and identify opportunities from our special situation to progress in research. Chance discovery is not serendipity, of making discoveries by chance, but the discovery of chance, or opportunity. In the normal sense of chance discovery as understood in the field, use is made of chance happenings of rare or significant events. The research is to discover such 'chance', and the methodology may include other research techniques such as data mining and granular computing.
We did experience a very rare (hopefully) event almost three years back in the December 26th Tsunami of 2004. However, there is no record of our making any use of this event to our benefit. I should of course record that there were very concerted efforts to make use of the displacement of a large number of people as an 'opportunity' to permanently banish them from their lands and to expropriate the beach, by the large scale organized tourist industry. This is of course another example of the dangers I referred to at the beginning. But we failed take to the chance to experiment with different and more benign forms of land settlement, rehabilitation with peoples' participation and work organization through workers' co-operatives.

My suggestion is about what I consider to be a much more significant 'event', rather than a 'rare' event such as the tsunami. It is about the special situation of the Sri Lankan researcher who is exposed to, and has access to two widely different cultures and traditions. Most of us have been born and bred into a culture where concepts such as

Going beyond logical argument
The relative and non-linear nature of time
The non existence of any permanent entity
Quantitative data being given qualitative characteristics
Exploring links between superficially disjointed facts, events etc.

are but second nature to us. On the other hand, we have had our formal training and education based on western classical scientific culture with its emphasis on reductionism.

I would wish to urge you to 'discover' this chance, given that it is likely to be invaluable in the pursuit of the emerging fields within AI.

I wish you and the Association success.

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Implementation of Fuzzy Controller for an Air Conditioning System

Gehan Anthonys, Hemali Pasqual, H. Sriyananda

Department of Electrical and Computer Engineering,
The Open University of Sri Lanka

E-mail: a.gehan@excite.com

Abstract

This paper presents the implementation of a Fuzzy Logic Controller for controlling an air conditioner system. The system has three functions: Cooler, Dehumidifier, and Air Freshener that would depend on the input parameters: Temperature, Humidity and CO_2 Air. The paper presents stages of analysis, design, simulation and implementation through the control parameters of the system.

Key words: Fuzzy Logic, Membership Function (MF), De-fuzzification

1. Introduction

This paper describes the design of a fuzzy logic controller for controlling an air conditioner system. It consists of three input parameters: Temperature, Humidity, CO_2, and three output parameters: Cooler, Dehumidifier, Air freshener. After initiating the system, the fuzzy controller controls the outputs according to the inputs.

The scope of this project spans from the investigation into fuzzy logic as applied to the Air conditioning system for constructing a real-time multi-output self-service system using a fuzzy logic controller.

This project aims to achieve the following:

- Reduce the design development cycle.
- Simplify the design complexity.
- Improve the control performance.

A prototype was built using the given design and tested for correct performance.

The rest of the paper is organized as follows: Section 2 describes the use of fuzzy logic controllers. Section 3 is on measuring the inputs and outputs of the system. Section 4 is on the design aspects. In Section 5 we report on the implementation of this prototype. Section 6 gives the experimental results.

2. Use of Fuzzy Logic Controllers

2.1. Fuzzy Sets and Subsets

Fuzzy logic is an extension to conventional (Boolean) logic. It can handle the concept of partial truth.
A fuzzy subset characterized by a membership function (MF) defined on a fuzzy set. An example of the subset "medium" of the set "Speed" is shown in Figure 2.1.

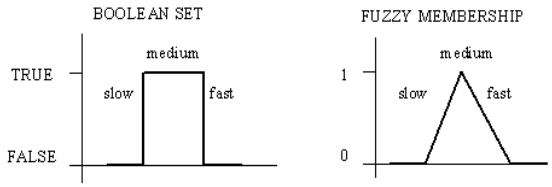


Figure 2.1: Boolean Sets versus Fuzzy Sets of a Parameter – Speed

The Fuzzy set in Figure 2.1 has a triangular shape. The set can be any shape including a trapezoidal one, or a bell curve. The triangular shape is used most often today because it is easy to work with and produces results similar to the more complex bell curve. When Fuzzy sets cover a whole range of possible values, they overlap so that a given value may be a member of more than one set. This makes a value unique.

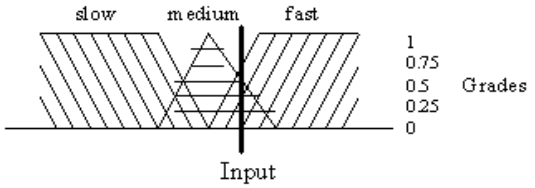


Figure 2.2: Overlapping Membership Set

A value indicated by the input line is a member of sets 'medium' and 'fast'. It has a grade of 0.75 in the 'medium' and 0.25 in the 'fast' sets. Therefore, these values are the fuzzy values, which process through the controller system.

2.2 Fuzzy controller

Fuzzy control originates from the human experiences. The objective of fuzzy logic control systems is to control complex process by using the experience of human beings. After many researchers' work, fuzzy control methods have changed from the original transition of human experience into control rule to a more engineering- oriented approach.

Conventional control systems use the physical models to deal with the process. That is time consuming and the designer needs a theoretical background. However, human being without any prior knowledge of the subject matter can control many processes. Therefore, formalization of the operator's experience is the main idea of fuzzy logic control by the methods of fuzzy logic.

2.3 Structure of a Fuzzy controller

Consider the basic fuzzy controller system as shown in Figure 2.3. It has several components and their functionalities are considered in briefly. Refer [6, 7].

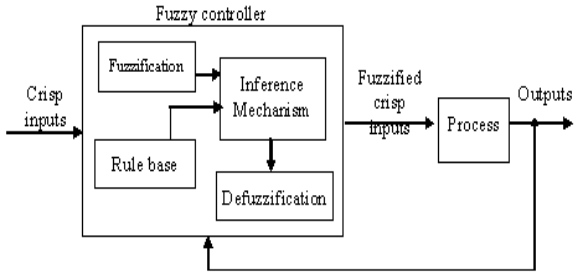


Figure 2.3: Components of a Fuzzy control system.

- **Rule Base** is a set of rules about how to control.
- **Fuzzification** is the process of transforming the crisp inputs (numeric inputs) into a form that can be used by the inference mechanism. (IM)
- **Inference Mechanism (IM)** uses information about the current inputs (formed by fuzzification), decides which rules apply in the current situation, and forms conclusions about what the process input should be.
- **Defuzzification** converts the conclusion reached by the IM into a numeric input for the process.

2.4 Fuzzy design process

First step in designing is to characterize the range of values for the input and output variables of the controller.

Then, assign labels for the parameters and write a set of simple English- like rules to control the system. If the designer does not tell the controller how to control the plant properly, it cannot be succeeded.
Inside the controller all parameters actions will be based on how the current parameters fall into these ranges and the rules describing the system behaviour. The controller's output will vary continuously to adjust the plant.

Therefore, in a fuzzy controller it accepts an input value, performs some calculations, and generates an output value. This process is called the Fuzzy Inference Process and works in three steps.

The fuzzy approach allows a control of non-linear systems without a deep knowledge of mathematical model. Imitating an expert's behavior easily generated fuzzy rules. However, membership functions for every input and output should be well defined and should be in complete domain.

3. Measurements

System provides two types of environments for the Inputs of the system. There are two ways to gather the data to the system: 'Auto Mode' and 'User Mode'.

In Auto mode, system senses input parameters by using sensors, while in User mode, system takes input values as well as default values for input parameters.

Finally, the results of the system are simulated in the computer screen and are implemented on the Prototype.

3.1 Data Acquisition

The data acquisition is doing either manually or by using sensors. The temperature sensor acts automatically while the other two sensor inputs are given using Push Buttons for this prototype. In this case, temperature sensor monitors the resistance with the change of temperature and it is programmed to convert its fuzzy values. Similar procedure is done for the others: Humidity and CO_2 sensors. (See Table 3.1)

Parameter	Sensing Device	Type
Temperature	Negative Temperature Control (NTC) Temperature sensor LM3914	Automatically
Humidity	Push Buttons	Manually
CO_2 Air	Push Buttons	Manually

Table 3.1: List of the Sensors in the system

4. Design Process

There is a specific flow to design the fuzzy controller. Figure 4.1 shows the flow chart for implementation of fuzzy controller.

4.1 Preprocessing

The inputs are most often hard or crisp measurements from some measuring equipment, rather than linguistic. The preprocessing is the first stage in the design process. It conditions the measurements before they enter the controller. By using many rules, it can make the controller multi dimensional, which makes it more difficult to design.

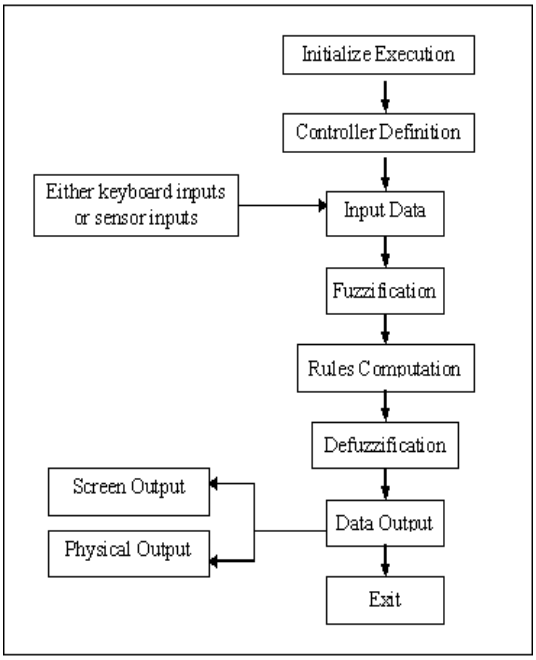


Figure 4.1: Sequence process of the System

4.2 Inference Processing

The rules of the system should be in a complete form. This ensures that for conditions of any process the controller can compute a meaningful control action. To build a complete rule system for a controller is feasible, though it would not be feasible in areas such as medicine where we are dealing with a number of unknown variables. In this case of fuzzy systems, namely a controller for a physical device, the variables have approximate (i.e. fuzzy) values but known.

In a number of cases, completeness is achieved by assuming all possible values for input (sensed) variables and considering all the combinations for the rules. Fuzzy controllers achieve this by abstracting the possible values in linguistic variables.

4.3 Defuzzification

Defuzzification is a process that converts a fuzzy set or fuzzy number into a crisp value or number. Defuzzification is such inverse transformation, which maps the output from the fuzzy domain back into the crisp domain. The following defuzzification methods are of practical importance.

- Center of Gravity (COG)
- Center of Maximum (COM)
- Mean of Maximum (MOM)
- Leftmost or Rightmost Maximum (LM or RM)

4.4 Experimental Setup

The system consist the following inputs and outputs with the fuzzy controller.

After sense the input by the sensors, they are processing in the fuzzy controller. Then controller returns the appropriate outputs to the control valve of the system. The following is the setup of the whole system.

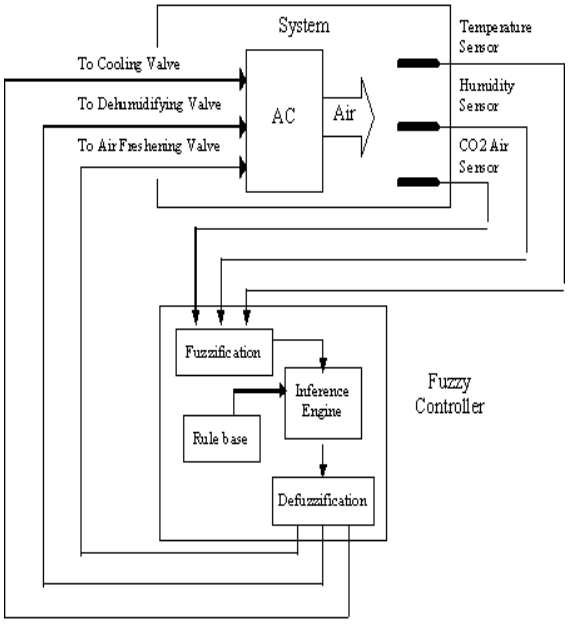


Figure 4.2: Block diagram of the System

5. System Implementation

The prototype and the PC interconnected by the parallel port of the PC.

5.1 Parallel port connections

A PC printer port is an inexpensive and yet powerful platform for implementing projects dealing with the control of real world peripherals. The printer port provides eight outputs, five inputs and four bidirectional leads and it provides a very simple means to use the PC interrupt structure.

Each printer port consists of three port addresses; data, status and control port. These addresses are in sequential order.

Data Register - This is used for sending data to the external devices.
Status Register - This is used to receive the data from the external devices.
Control Register - This is used to send the data to control the external devices.

5.2 Pin connection of the System

Figure 5.1 shows the Pin assignments of the system with the Parallel port of the PC. This represents the Registers' configuration for each parameter to the PC using its Parallel port.

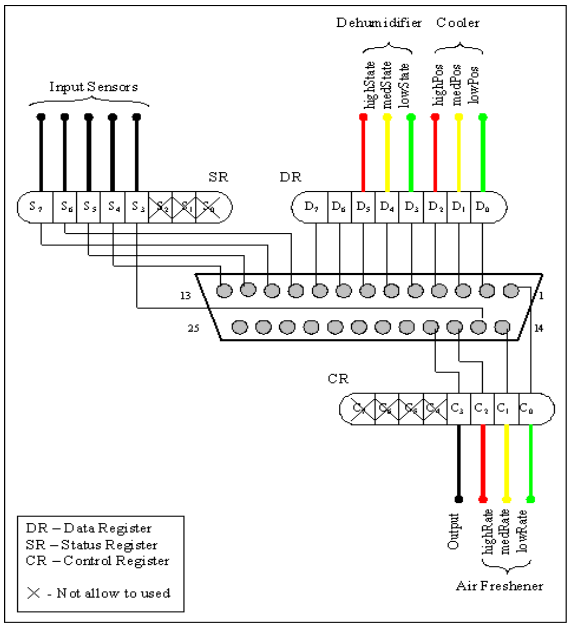


Figure 5.1: DB25 female connector of the system

6. Simulation Results

System particularly interested in the following effects to the system:

- Variations of the Temperature/Humidity/CO_2 Air with the time
- Comparison of the variation of three input parameters
- Variation actions of the Cooler/ Dehumidifier/

Air Freshener with the time

- Comparison of the actions of the three control Parameters

The appropriate fuzzy values for each input parameter and each control parameter are simulated in a graphical manner of the system.

Using the above results the progress of the system can be simulated. The above effects are simulated in 2D graphs in the system.

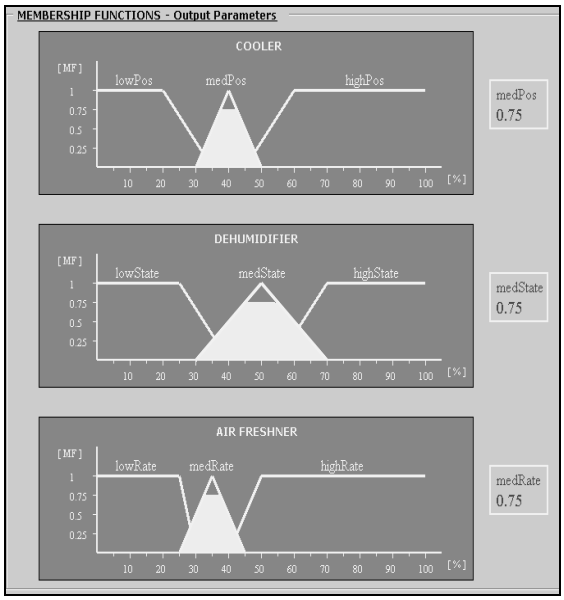
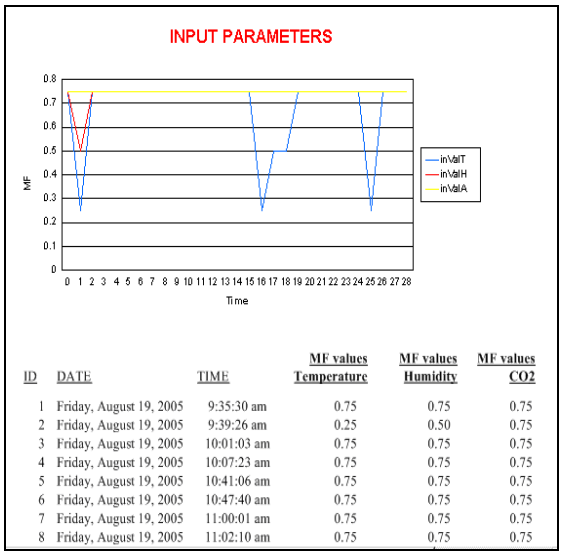
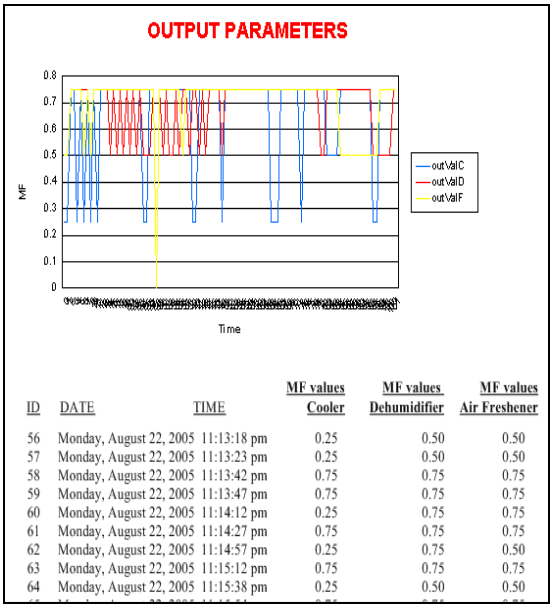


Figure 6.1: MFs for the Output Parameters



ID	DATE	TIME	MF values Temperature	MF values Humidity	MF values CO2
1	Friday, August 19, 2005	9:35:30 am	0.75	0.75	0.75
2	Friday, August 19, 2005	9:39:26 am	0.25	0.50	0.75
3	Friday, August 19, 2005	10:01:03 am	0.75	0.75	0.75
4	Friday, August 19, 2005	10:07:23 am	0.75	0.75	0.75
5	Friday, August 19, 2005	10:41:06 am	0.75	0.75	0.75
6	Friday, August 19, 2005	10:47:40 am	0.75	0.75	0.75
7	Friday, August 19, 2005	11:00:01 am	0.75	0.75	0.75
8	Friday, August 19, 2005	11:02:10 am	0.75	0.75	0.75

Figure 6.2: Variation of three input MF values with time



ID	DATE	TIME	MF values Cooler	MF values Dehumidifier	MF values Air Freshener
56	Monday, August 22, 2005	11:13:18 pm	0.25	0.50	0.50
57	Monday, August 22, 2005	11:13:23 pm	0.25	0.50	0.50
58	Monday, August 22, 2005	11:13:42 pm	0.75	0.75	0.75
59	Monday, August 22, 2005	11:13:47 pm	0.75	0.75	0.75
60	Monday, August 22, 2005	11:14:12 pm	0.25	0.75	0.75
61	Monday, August 22, 2005	11:14:27 pm	0.75	0.75	0.75
62	Monday, August 22, 2005	11:14:57 pm	0.25	0.75	0.50
63	Monday, August 22, 2005	11:15:12 pm	0.75	0.75	0.75
64	Monday, August 22, 2005	11:15:38 pm	0.25	0.50	0.50

Figure 6.3: Variation of three output MF values with time

7. Conclusion and Future Improvements

Fuzzy logic has found several practical applications in industrial control. Fuzzy logic control can be viewed as a development of the "expert systems" concept but with the logic based on fuzzy sets. Fuzzy logic allows quantitative information to be incorporate into a framework based on "if then" rules. This provides a powerful means for expressing the objectives of a control system and implementing these objectives.

This document has described the phases of the implementation of a fuzzy logic controller for an industrial application. The project has involved examining the fuzzy controller for an Air conditioning system control.

The system response was studied for three input parameters and three output parameters to control the system. Fuzzy logic control concepts were used for the design.

The result of the controller is analyzed by simulating results with the suitable graphical representation and it is useful to improve the performance of the system.

At present, the system considers five, three and three crisp values for the input parameters Temperature, Humidity and CO_2 respectively. Therefore, it is better to extend the number of crisp values for each parameter in both Input and Control Parameters.

8. References

[1] George J. Klir, Bo Yuan "Fuzzy Sets and Fuzzy Logic – *Theory and Applications* pp. 327–356, 1997; ISBN: 81-203-1136-1

[2] Junhong Nie, Derek A. Linkens "Fuzzy-Neural Control – *Principles, Algorithms and Applications*, 1998; ISBN: 81-203-1368-2

[3] Bart Kosko "Neural Networks and Fuzzy Systems" Englewood Cliffs, NJ: Prentice Hall, pp. 379–406 and pp. 407–424, 1995; ISBN: 81-203-0868-9

[4] Toshiro Terano, Kiyoji Asai, Michio Sugeno "Fuzzy Systems Theory and Its Applications, Academic Press, Inc. pp. 125-136, 1991; ISBN: 0-12-685245-6

[5] James P. Ignizio "Introduction to Expert Systems – The Development and Implementation of the Rule - Based Expert Systems, McGraw-Hall Inc., 1991

[6] Tariq Samad, "Perspectives in Control Engineering *Technologies, Applications and New Directions*, IEEE Control systems Society, pp. 104-133, 2001; ISBN: 0- 7803-5356-0.

[7] James A. Momoh, Mohamed E. El-Hawary "Electric Systems, Dynamics and Stability with Artificial Intelligence Applications, Marcel Dekker, Inc. pp. 175 - 220, 2000; ISBN: 0-8247-0233-6

[8] I. Trouch "Mathematical and Computer Modeling of Dynamics Systems – *Methods, Tools and Application in Engineering and Related Sciences*", Swets and Zeitlinger, vol. 7, no. 1, pp. 20-30, March 2001; ISBN: 1387-3954

[9] http://www.aptronix.com

[10] http://www.iau.dtu.dk

[11] http://www.austinlinks.com/fuzzy/

[12] http://www.omega.com/techref/fuzzylogic.html

[13] http://www.comp.ai.fuzzy

[14] http://www.etse.urv.es/~aoller/fuzzy/fzzlog.doc

[15] http://pcc.imir.agh.edu.pl/

[16] http://ceaspub.eas.asu.edu/

[17] http://www.electronic-engineering.ch/microchip/

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Using Human-Assisted Machine Translation to overcome language barrier in Sri Lanka

B. Hettige[1], A. S. Karunananda[2]

[1]Department of Statistics and Computer Science, Faculty of Applied Science, University of Sri Jayewardenepura, Sri Lanka.

[2]Faculty of Information Technology, University of Moratuwa, Sri Lanka.
budditha@yahoo.com[1] , asoka@itfac.mrt.ac.lk[2]

Abstract

Automated machine translation has faced with many issues regarding handling of semantics. In general, this issue can be addressed by computer-assisted machine translation at the pre-editing and post-editing stages. Our research has gone further and introduced an intermediate editing stage just before morphological analyzer of the target language. This approach detects semantic issues before hand and allows addressing those by human intervention. As a result the final translation will be more realistic and cut down the need for human intervention at the post-editing stage. The above approach has been used to develop English to Sinhala machine translation system. The system has been developed using Prolog and Java to run on a standard PC.

1. Introduction

Language barrier has been a major cause for not being able to disseminate world knowledge for rural communities those who do not use English as the mother tongue. The obvious solution to this issue is the use of modern computing technologies to translate from English to local languages. Many European and Asian countries have already taken steps to develop machine translation systems. In the Asian region, Indians have developed a variety of machine translation systems, including Mantra [5], Matra, Anusaaraka [2], Angalabharthi [4] and Angalahindi [3]. Among others, EDR [15] by Japanese is one of the most successful machine translation systems in the world. These translation systems use various approaches to machine translation, including, Human-Assisted Translation, Rule based Translation, Statistical Translation, Example-based and Knowledge-based Translation. However, due to various reasons associated with complexity of languages, for more than last fifty five years, Machine Translation (MT) has been identified as one of the least achieved area in computing. Most of these issues are associated with semantic handling in MT systems. Obviously, one approach to solve this issue is the use of post-editing[12] by humans. However, we argue that tedious work at the post-editing stage can be reduced by introducing an intermediate-editing stage just before running the morphological analyzer of the target language. This is mainly because; the intermediate-editing can ensure that only the appropriate words of the target language are sent forward.

With the above philosophy, we have been working on the development of English to Sinhala machine translation system. As per this system, we have already developed the Sinhala parser [6], Sinhala morphological analyzer [7], Transliteration module [9] and seven dictionaries [10]. The Sinhala parser and morphological analyzer have been tested through various applications such as Sinhala Chatbot [8]. This paper presents our approach to intermediate-editing as an expansion to the on going machine translation project. The system has been developed using Prolog [13] and Java to run on a standard PC.

The rest of this paper is organized as follows. Section 2 describes the overview of some existing Machine Translation systems. Section 3 describes design of the Human-Assisted English to Sinhala Machine Translation System. Section 4 presents how system works. Finally, Section 5 concludes the paper with a note on further work.

2. Some Existing MT Systems

Machine Translation (MT) is a translation process that translates one natural language into another [12]. In general, any machine translation system contains a source language morphological analyzer, a source language parser, translator, target language morphological analyzer, target language parser and several lexicon dictionaries. Source language Morphological analyzer analyzes a Source language word and provides Morphological information. Source language parser is a syntax analyzer that analyzes source language sentence. Translator is used to translate a source language word into target language. Target language Morphological analyzer works as a generator and it generates appropriate target language words for given grammatical information. Also target language parser works as a composer and it composes a suitable target language sentence. Further more, any Machine Translation system needs minimum of three dictionaries such as the source language dictionary, the bilingual dictionary and the target language dictionary. Source language morphological analyzer needs a source language dictionary for Morphological analysis. Bilingual dictionary is used by the Translator for translating source language into target language; and the target language morphological generator uses the target language dictionary to generate target language words. Regarding English to Sinhala machine translation point of view, the Machine Translation system needs an English dictionary, an English-Sinhala bilingual dictionary and a Sinhala dictionary.

Machine translation (MT) is a complex and a difficult task. However a large number of MT Systems have been developed for many languages all over the world. Sinhala is an Indo Arian language and Some Indian languages such as Pali, Sanskrith and Tamil are closer to Sinhala language. Therefore we need to study some existing MT systems especially the ones developed for Indian languages. At present Indians has developed a variety of machine translation systems. Below is a brief description on them.

The Anusaaraka [2] is a popular machine-aided translation system for Indian languages that makes text in one Indian language accessible to another Indian language. Also this System uses Paninian Grammar (PG) model [1] to its language analysis. The Anusaaraka project has been developed to translate Punjabi, Bengali, Telugu, Kannada and Marathi language into Hindi. The approach and lexicon is general, but the system has mainly been applied for children's stories.

MaTra [5] is yet another Human-Assisted translation system for translating English to Indian languages. This approach uses a 'tag' system to represent grammatical information of the language at hand. MaTra has been developed for the domain of gazette notifications pertaining to government appointments [5].

Angalabharti [4] is also human-aided machine translation system used in India. Since India has many languages, there are a variety of machine translation systems. For example, Angalahindi [3] translates English to Hindi using machine-aided translation methodology. Human-aided machine translation approach is a common feature of most Indian MT systems. In addition, these systems also use the concepts of both pre-editing [15] and post-editing[12 as the means of human intervention in the machine translation system.

Among others, Electronic Dictionary Research (EDR) [16] is the most successful machine translation system. This system has taken a knowledge-base approach in which the translation process is supported by several dictionaries and a huge corpus. While using the knowledge-based approach, EDR is governed by a process of statistical MT. As compared with other MT systems, EDR is more than a mere translation system but provides lots of related information. This ability has been possible since EDR has a huge reservoir of knowledge in the form of corpus and dictionaries. For instance, EDR has a word dictionary, concept classification dictionary; concept description dictionary; co-occurrence dictionary and bilingual dictionary. Due its rich reservoir of linguistic knowledge EDR has become a considerably automatic translation system. Therefore, in comparison with

most Indian MT systems, human-aided machine translation is not necessarily encouraged in EDR. It is evident from the discussion and human-aided machine translation is more practical to consider for MT projects which are at their early stages. Therefore, the proposed English to Sinhala MT system has also taken the approach of human-assisted translation. However, we go beyond pre-editing and post-editing, and introduce an intermediate-editing stage to a MT system.

3. Design of English to Sinhala MT System

We have designed the proposed English to Sinhala MT system with seven modules, namely, English Morphological analyzer, English parser, Translator, Sinhala Morphological analyzer, Sinhala Parser, Transliteration module and Lexical dictionaries. Figure 1 shows Design of the English to Sinhala MT System. Brief descriptions of each component are given bellow.

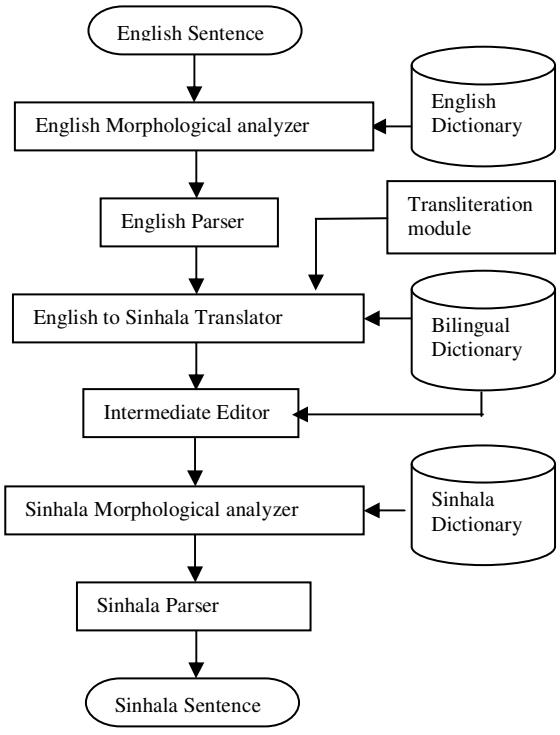


Fig 1: Design of the English to Sinhala Machine Translation System

3.1 English Morphological analyzer

English Morphological analyzer reads a given English sentence word by word and identifies morphological information for each word. There are many Morphological analyzers available for English language. Therefore, in this development, we have customized an existing English morphological analyzer. At this stage of the project, we assume that the sentences input to the MT system, has no spelling and grammatical mistakes. As such we can use a simple morphological analyzer for the English language at this stage. The morphological analyzer in our MT system has linked up with an English dictionary to get grammatical information of the words in the input sentence. SWI-Prolog has been used to implement this morphological analyzer.

3.2 English Parser

English parser receives source English sentences and tokens from English Morphological analyzer. This parser works as a syntax analyzer. Since there are many English parsers, we have customized an existing parser for our purpose. The current version of the parser used in our MT system mainly concerns only about the simple sentences. The parser has also been implemented using SWI-PROLOG.

3.3 Translator

Translator is used to translate English base word into Sinhala base word with the help of bilingual dictionary. This translator is a simple one and it does not automatically handle semantic of sentences. We argue that this stage can be supported by human intervention to generate the most appropriate translation for some words in a sentence. As such handling semantic, pragmatic and Multiword expressions must be addressed with the support from humans, for which we introduce and intermediate-editor.

3.4 Intermediate-Editor

English to Sinhala Human-Assisted Machine Translation system uses Intermediate Editor to handle ambiguities in semantic, pragmatic and Multiword expressions before proceeding to Sinhala linguistic modules in the MT system. Intermediate Editing facility is provided as a

human interface for the MT system. This editor provides with facilities such as showing synonyms, anti-synonyms, related words, etc. The intermediate-editor is linked up both English and Sinhala dictionaries in the MT system. The process of intermediate-editing, before composing a Sinhala sentence, drastically reduces computational cost for running Sinhala morphological analyzer and parser. In addition, requirement for post-editing can be reduced by the process of intermediate editing. On the other hand, intermediate-editing can be used as means of continuous capturing of human expertise for machine translation. This knowledge can be reused for subsequent translations. As such the concept of intermediate-editing can be introduced as an approach to automatic knowledge management in a MT system. It should be noted that the knowledge used for pre-editing and post-editing cannot be readily captured by a MT system, as these process can be done even out side MT system. In contrast, intermediate-editing will be a integral part of the MT system, in which human directly interact with the system. The intermediate-editor of the MT system is a Java-based implementation.

3.5 Sinhala Morphological Analyzer

The Sinhala Morphological analyzer [7] works as a Morphological generator. This Morphological analyzer reads the words from Translator (as improved by a human when necessary) word by word. For each word, the morphological analyzer generates the appropriate word with full grammatical information such as nama (nouns), kriya (verb) and nipatha (preposition) in Sinhala language. This morphological analyzer works with the help of three dictionaries, namely, Sinhala Rule dictionary, Sinhala Word dictionary and Sinhala Concepts dictionary. All these databases and the morphological analyzer are implemented using Prolog.

3.6 Sinhala Parser

The Sinhala parser [6] works as a Sentence composer. It receives tokenized words from the morphological analyzer and composes grammatically correct Sinhala sentence. In generally, a Sinhala sentence contains 5 components, namely Ukktha vishashana (adjunct of subject), Ukkthya (Subject), karma vishashanaya (attributive adjunct of object), karmaya (object) and akkyanaya [8]. These five components of a Sinhala sentence are the building blocks for design and implementation of a Sinhala parser. The parser is also one of the key modules of this Human-Assisted English to Sinhala Machine Translation System and it is also implemented using SWI-PROLOG.

3.7 Dictionaries

Translation system uses six dictionaries such as English word dictionary, English concepts dictionary, English-Sinhala bilingual dictionary, Sinhala word dictionary, Sinhala rule dictionary and Sinhala concept dictionary [10]. English word dictionary contains English words and the lexical information. English concept dictionary contains synonyms, anti-synonyms and general knowledge about English words. English to Sinhala bilingual dictionary is used to identify appropriate Sinhala base word for a given English word and it contains relation between English and Sinhala words. Sinhala word dictionary stores Sinhala regular base words and lexical information. Same as English dictionary, Sinhala Concept dictionary stores Symantec information. The Sinhala rule dictionary stores rules required to generate various word forms. These are the inflection rules for formation of various forms of verbs and nouns from their base words. The rule dictionary also stores vowels, consonants, upasarga (prefix) and vibakthi (postfix).

3.8 Transliteration module

MT system needs to solve Out-of-vocabulary problems and handle technical terms. Machine transliteration can be used as a resalable solution for that. Transliteration is the practice of transcribing a word or text written in one writing system into another writing system [12]. In other words, Machine transliteration is a method for automatic conversion of words in one language into phonetically equivalent ones in another language. At present we have developed two types of transliteration models. One of these models transliterates Original English text into Sinhala Transliteration and the other transliterate Sinhala words that are written in English which transliterate into Sinhala. Finite State transducers are used to develop these two modules [9]

4. How System works

In this section we describe how translation system works for a given input sentence. For example, assume that the system reads "Saman eats red rice for his lunch" as the input sentence. Then the English Morphological analyzer identifies each word and returns the following Prolog predicates.

unknown([un001], 'Saman').

everb([ev01], 'eats').

eadj([ea01], 'red').

enoun([en02], 'rice').

eprep([ep01], 'for').

epnoun([en03], 'his').

enoun([en04], 'lunch').

Now the English Parser reads the original English sentence together with the output of the Morphological analyzer. After this analysis, the parser returns the following information.

subject ([un001]).

object([ea01, en02]).

objectp([ep01, en03, en04]).

verb([ev01]).

Tokenized ID of English words are then forwarded to the translator. The translator identifies Sinhala base word for each English word in the sentence, with the help of bilingual dictionary. It should be noted that, the first word 'Saman' is an unknown word in the dictionary. Therefore, it is out-of-vocabulary and translator cannot translate the word. As a result, the translator uses Transliteration module to get an appropriate Sinhala Transliteration. Then the output of the translator is as follows.

spronoun(un001, 'සමන්').

sverb([sv01], 'කනවා').

snoun([sn02], 'රතු').

snoun([sn03], 'බත්').

snoun([sp01], 'සඳහා').

snoun([sn04], 'ඔහුගේ').

snoun([sn05], 'දිවා ආහාරය').

At this point human can change the above translated output by using Intermediate-editor. For example, the English word 'rice' contains several Sinhala meanings such as 'ගොයම්', 'සහල්', 'වී', 'බත්', 'හැල්' etc. Now human can select the most suitable Sinhala word for the word 'rice'. Also the English word 'for' has several meanings such as 'ට', 'සඳහා', 'වෙනුවට', 'ගැන' 'නිසා', 'පිණිස' etc. Through human intervention the word 'සඳහා' can be selected. After that, the Sinhala Morphological analyzer reads all these words and generates appropriate Sinhala words with grammatical information. Output of the Sinhala Morphological analyzer is as follows.

snoun([sn01], ,3,1,3,0,1,v1, 'සමන්').

sadjn([sn04], 'දිවා').

snoun([sn01], 3,1,3,0,1,v2, 'ආහාරය ').

sprep([sn03], 'සඳහා').

sadjn([sn04], 'රතු').

snoun([sn04], 3,1,3,0,1,v5, 'බත්',…).

sverb([sv01] 3,1,3,0,1,1, 'කයි').

All the above information are sent forward and reserved by Sinhala parser. The Sinhala Parser identifies the following linguistic details in Sinhala language and it generates appropriate Sinhala Sentence.

Subject ('සමන්').

Object (රතු බත්).

Objectp('දිවා ආහාරය සඳහා').

Verb(කයි).

As the last step of the translation process, Sinhala parser composes the corresponding Sinhala sentence 'සමන් දිවා ආහාරය සඳහා රතු බත් කයි'. How System translate the given

sentence, by using Intermediate editor is described bellow. Figure 2 shows a user Interface of the English to Sinhala Machine Translation System.

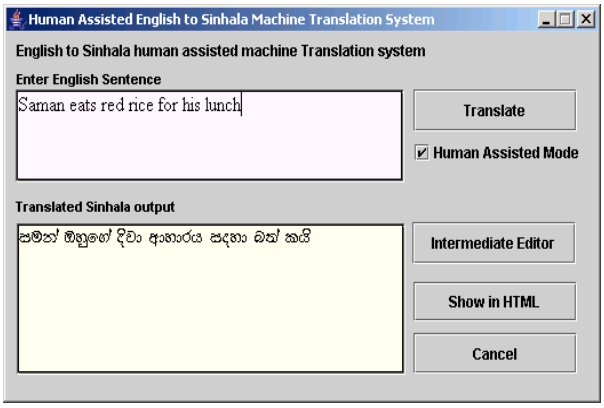


Fig. 2: User Interface of the English to Sinhala Machine Translation system

After Starting the Translation process the system automatically shows Intermediate Editor for selecting the suitable words. Using this editor, assistant can easily select the most suitable word. Figure 3 shows Intermediate Editor.

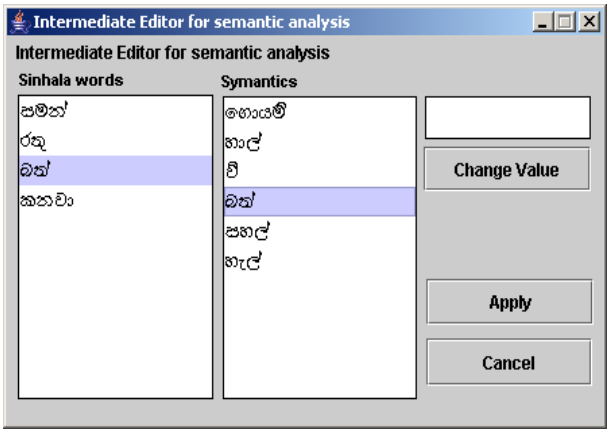


Fig 3: Intermediate Editor for Symantec analysis

It shows numbers of synonyms for the word 'බත්'. Note that, Sinhala Morphological analyzer needs appropriate Sinhala base word and all the required grammatical information to generate Sinhala words [8]. By using this intermediate editor the user can select appropriate base word. However, some grammatical information does not need to change. These information are automatically generated by the previous stages such as Translation, Sentence and word analysis.

5. Conclusion and Further Works

Our objective of this project was to develop English to Sinhala Translation System. In this sense, we have designed and developed a Human-Assisted English to Sinhala translation system with a particular emphasis on an intermediate-editing through human intervention. This approach has reduced the workload at the post-editing stage of human-assisted machine translation. In addition it brings higher level of accuracy towards a meaningful translation.

Improvements and expansions to dictionaries will be an essential further work of this project. In addition, we intend to develop a capacity for the system to learn from intermediate-editing results to enable evolution of the translation system towards an automated system.

6. References

[1] Askhar B, Chaitanya V, Sangal R, "Natural Language Processing: A Paninian Perspective", Prentice Hall of India, New Delhi, India, 1995.

[2] Bharathi A, Chaitanya V, Kulkarni A. P, Sangal R., "Anusaaraka: Overcoming language barrier in India", to appear in "Anuvad: Approaches to Translation", Rukmini Bhaya Nair, (editor), Sage, New Delhi, 2001.

[3] Sinha R.M.K, Jain A., "AnglaHindi: an English to Hindi machine-aided translation system", MT Summit IX, New Orleans, USA, 23-27 September 2003; pp.494-497.

[4] Sinha R.M.K, "Integrating CAT and MT in AnglaBhart-II architecture", 10th EAMT conference, May. 2005, pp. 235-244.

[5] Durgesh R., "Machine Translation in India: A Brief Survey", National Centre for Software Technology, Mumbai, India. http://www.elda.org/en/proj/scalla/scalla2001/scalla2001Rao.pdf

[6] Hettige B., Karunananda A. S., "A Parser for Sinhala Language – First Step Towards English to Sihala Machine Translation", To appear in the proceedings of International Conference on Industrial and Information Systems(ICIIS2006), IEEE, Sri Lanka, 2006.

[7] Hettige B., Karunananda A. S., "A Morphological analyzer to enable English to Sinhala Machine Translation", Proceedings of the 2nd International Conference on Information and Automation (ICIA2006), Colombo, Sri Lanka, 2006 pp. 21-26.

[8] Hettige B., Karunananda A. S., "First Sinhala chatbot in action", Proceedings of the 3rd Annual Sessions of Sri Lanka Association for Artificial Intelligence(SLAAI), University of Moratuwa, 2006.

[9] Hettige B., Karunananda A. S., "Transliteration System for English to Sinhala Machine Translation", proceedings of second International Conference on Industrial and Information Systems(ICIIS2007), IEEE, Sri Lanka, 2007.

[10]Hettige B., Karunananda A. S., "Developing Lexicon Databases for English to Sinhala Machine Translation", proceedings of second International Conference on Industrial and Information Systems(ICIIS2007), IEEE, Sri Lanka, 2007.

[11] A. M. Gunasekara, A Comprehensive Grammar of the Sinhalese Language", Asian Educational Services, New Delhi, Madras, India., 1999.

[12] Jeff L., Christopher H, "Toward the development of a post-editing module for Machine Translation raw output", Presented at the Third International Controlled Language Applications Workshop (CLAW2000), Washington, 2000.

[13] wikipedia, the free encyclopedia, http://en.wikipedia.org.

[14] SWI-PROLOG home page; http://www.swi-prolog.org

[15] Toshio Y., "Improvement of Translation Quality of English Newspaper Headlines by Automatic Preediting", in Proceedings of the MT Summit VII, 1999.

[16] Toshio Y, "The EDR electronic dictionary", Communications of the ACM, Volume 38, Issue 11 (Nov. 1995), pp. 42 – 44.

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

On Computing Ontology for Mental Factors

[1] Subha D. Fernando, [2]Asoka S. Karunananda

[1, 2] Faculty of Information Technology, University of Moratuwa, Moratuwa, Sri Lanka
{[1]subha, [2]asoka}@itfac.mrt.ac.lk

Abstract

Ontology has introduced a new paradigm for software development. Nowadays, Ontologies has become a key theme of areas like Semantic Web, Web Services and Multi Agent Systems. However, a few research projects are carried out to develop a comprehensive Ontology for modeling of mental factors. This has been a barrier for modeling of computer systems concerning human emotions and sensations. We have exploited Buddhist theory of mental factors and developed a Mental Ontology environment to model systems involving mental factors. Our implementation introduces a base class mental factor with various feature and behaviors as per Buddhist theory of mind. This base class can be extended to create an arbitrary mental factor. The paper also explains how the developed mental Ontology environment can be used by an Ontology developer in Multi Agent System.

Keywords: Ontology, Semantic Web, Agent Based Systems, Mental Factors.

1. Introduction

For many centuries, Ontology has been recognized as the artifact for enabling communication among humans. With the ever increasing growth of the World Wide Web, people are now interested in machine-machine communication with limited or no human intervention. As such, many researches are involved in the construction of Ontolgies to enable interoperability of the resources available in various data formats running on heterogeneous hardware and software environments on the next generation of the World Wide Web, known as the Semantic Web. Some application areas of Ontology include knowledge management [1], automated data integration and maintenance in web portals [2], e-learning [3] and e-commerce [4]. These applications necessarily involve manipulation and sharing of information among various sources distributed over complex networks. At present, such systems are developed by looking at the data and knowledge sources from a very rigid view that describes the world through objects, entities, relations, attributes, etc. However, no such systems are generally capable of capturing mental attributes such as human, interest, desire, dissatisfaction, fear, etc. to make Ontologies more comprehensive and meaningful, especially when human being are involved in transactions.

Currently a few researches have been conducted to model Ontology for modeling of emotions. These researchers concentrate on building of Ontologies merely through physical means such as facial expressions without considering of mental attributes [5]. However, researchers have identified the necessity of developing a comprehensive computing Ontology for emulating emotions on machines [6]. Undoubtedly, such research will expand horizons of many areas including humanoid robots, unman vehicles, multi agent systems, man-machine co-existence systems and many other agents software that directly communicate with humans.

This paper presents our exploitation of Buddhist theory of mental factors and the construction of Ontology for mental factors that can be used for design and development of applications concerning emotions. The rest of the paper is organized as follows.

Section 2 provides with an overview of the current research in the area of Ontology. Section

3 describes the Buddhist theory of mental factors. Section 4 reports on the design, the implementation and the application of Mental Ontology. Section 5 concludes the paper with a discussion of novelty and further work of this research.

2. Overview of Ontological Modeling

Although there is no general definition to the Ontology, it is commonly accepted as a model, where great philosophers and computer scientists try to implement the reality of the world. Philosophers define Ontology in terms of concepts and their relationships. However, computer scientists, still prefer to have object oriented way of defining it in terms of classes. Therefore in computer paradigm, Ontology consist of classes of objects, such as chair, desk, truck , customer, supplier, etc , classes of attributes like color, idnumber, material type, etc and classes of behaviors which specify allowable intercommunication and privilege actions corresponds to the objects. Ontological modeling had gained advantage by separating its structural representation from actual implementation.

Since there is no general agreement on the Ontology definition, some computer scientists look at Ontological modeling from object oriented aspect. Even though there is a similar structure for both object oriented modeling and Ontological modeling, we see a great difference in evaluation procedures. For an example, at the evaluation process, object oriented evaluates the systems in terms of transactions and object interactions where as ontology evaluates the system in terms of knowledge representation and knowledge evaluation.

However, Gruber's definition on Ontology [15], "*an explicit specification of a conceptualization"* has become the frequently used definition in computing. According to the Buddhist philosophical view, conceptualization or the perception of a particular domain depends on the physical and mental states of a viewer. This has been proven by the current research where Ontology building has now become a collaborative task of different domain experts and stakeholders. Even though this can be taken as a solution at the implementation level, evolution and emergence of new knowledge in Ontology is still need to be considered.

In the next section, we explain some ongoing major researches in Ontological modeling.

2.1 Major Research Areas in Ontology

Researches on Ontology based modeling have been carried out with different perspectives by different domains expertise. The underground motivation of all these researches is either knowledge representation, knowledge process and knowledge evolution. Review of this section is limited to two key research areas, namely Semantic Web and Agent Based systems, where knowledge representation, processing and evolution are crucial.

Semantic web is an extension to the current Web, which attempt to enable the machine-machine communication with or without human intervention. According to literature, Semantic Web research has two main directions, namely, (a) converting existing web pages into machine-operatble format and, (b) creating new web pages with machine understandable format.

Multi Agent Systems is yet another area where the concept of Ontology can play an immense role in supporting communication and sharing problem solving knowledge among related entities. Many researchers have cited the value of Ontology as an artifact for supporting over all functionality of a Multi Agent System. Many systems including aircraft maintenance, logistic planning [7] and industrial automation [8] have shown the potential of the use of Multi Agent Systems Technology from an Ontological viewpoint.

Web Service [10] and Knowledge Management [1] are also some other areas where Ontological modeling has been successfully applied.

At present there are hundreds of tools and software environments for development of Ontology based systems. Among others, OntoWeb [9] research group has done critical review of these tools and has evaluated all these tools in terms of interoperability, knowledge representation, inferences services attached and usability. For their evaluation, tools have been categorized as (a) Ontology development tools, (b) Ontology merge and integration tools (c) Ontology evaluation tools (d) Ontology based annotation tools (e) Ontology storage and

querying tools (f) Ontology learning tools. According to their review, neither these tools interoperate and nor they covered the life cycle of the ontology development. Karunananda and Rzevski [11] have also pointed out that these tools are unable to model Ontology evolution, which is an essential feature of Ontolgoies in many domains.

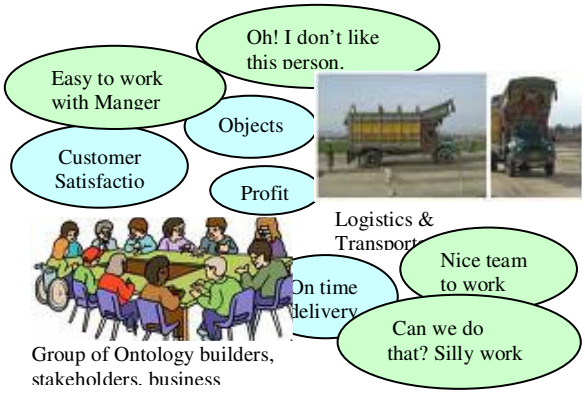


Figure 1: Ontology Building Environment.

In fact, Karunananda and Rzevski [11] have identified two major issues in Ontology development; (a) Currently Ontology developers usually implement their own worldview. (b) Ontology is only constructed as permanent structures without considering the emergence or evolution of knowledge. In their research, they have identified the necessity of concerning mental aspects in Ontology building (Figure 1).

As per Figure 1, usually any environment can be characterized by two main attributes, namely physical attributes, related to perception of particular domain, and mental attributes, related to the mental states of each ontology builders, stakeholders, etc. Therefore success of the task depends on the interactivity among group members. The Ontology is the device for enabling such interactions.

High interactivity among members guarantees the knowledge evolution and emergence. Nevertheless, this interactivity depends on the personal view about other group members. Sometimes Personal Manager (PA) will like to work with Developer (D) than Assistance Manager (AM). In that case, the productivity is high when PA interacts with D than AM. Therefore, it is necessary to implement mental states of each agent or any other entities, which involve with Ontology, then it, will guarantee the evolution and emergent of knowledge in any environment.

2.2 Research in Mental Ontology

Over many years most of the Ontology related research have focused on knowledge engineering and knowledge processing in terms of physical attributes in respective domains, but few research have been done in relation to development of mental Ontologies.

In the recent past, Kansei Engineering has emerged as a method of translating feelings and expressions into product parameters. Using this technology Japanese has raised a project called Kansei, which contains modeling and implementation of cognitive science as a one part of the project [12].

Research has also been conducted to build Buddhist Ontology of mental factors for Kansei systems [6]. This study has identified Buddhist classification of mental factors and their relationships. Further, a Buddhist mental Ontology pertaining to humanization of e-learning environments has also been developed [13]. These works primarily consider a sub set of mental Ontologies describes in Buddhist Philosophy. As such, development of full pledge comprehensive mental Ontology remains as a research challenge.

Karunananda and Rzevski [11] have identified full spectrum of mental Ontology in Buddhism. They have also explained how a computer model of this particular Ontology can be constructed. In their XML description, it was explained how each agent can be defined in terms of physical and mental attributes. For example, mental factors such as satisfaction, motivation, confidence, can be incorporated together with the same physical attributes such as agency, agent_id, and task. However, that work has not yet developed a computer model of mental Ontology.

Our research has taken steps to develop a full scale general purpose mental ontology as defined in Buddhist Philosophy.

3. Buddhist Ontology

The Buddhist philosophy analyzes the mental and physical phenomena of all beings under Abhidhamma Pitaka [14]. Therefore, Buddhist Ontology of the world describes both physical and the mental aspects. According to Buddhism, the mental phenomenon is sub divided as mental factors and consciousness. In fact, consciousness themselves are some valid combinations of mental factors [14]. As such mental factors characterize a given consciousness. Buddhism defines eighty nine distinct consciousnesses that are formed by fifty two different mental factors. Since mental factors work as the basis for consciousness too, the study of mental factors is fundamental to modeling of a Mental Ontology.

3.1 Classification of Mental Factors

The Buddhist philosophy has identified fifty two mental factors that characterize thoughts or consciousness. In other words, each consciousness can be seen as a combination of mental factors. Buddhism classifies these 52 mental factors into four categories, namely, universals, particulars, unwholesome and wholesome [14].

Seven mental factors: Contact, Feelings, One pointed ness, perception, Psychic Life and Attention appear in all consciousness, and are called Universal. According to the Buddhist philosophy the consciousness with only seven mantel factors cannot be occurred and at least human being needs thirteen (13) mental factors to form the simplest possible consciousness. Other six essential mental factors are called particulars, namely, Initial Application, Sustained Application, Deciding, Interest, Effort and Intention.

Remaining thirty-nine (39) mental factors are categorized as unwholesome and wholesome mental factors. There are 14 unwholesome mental factors that form the consciousness pertaining to bad thoughts and actions. In contrast, 25 wholesome mental factors form consciousness regarding good actions and thoughts.

Buddhism has identified not only different class of mental factors, but also the relationship among these mental factors. In depth study of such relationships are essential for the development of a Mental Ontology.

3.2 Relationships among Mental Factors

A mental factor alone doesn't provide any meaning to the consciousness, until it is associated with other mental factors. Therefore, it is necessary to identify the valid combination of mental factors, which characterizes the consciousnesses. The Buddhist philosophy has precisely defined the relationships among mental factors as a set of rules. As such mental factors categorized under universal category, should occur in all the states of consciousness, while mental factors categorized under unwholesome category can occur in only in unwholesome states of consciousness. Further, it classifies unwholesome mental factors namely dullness, lack of moral shame, lack of moral fear and restlessness should occur in each unwholesome states of consciousness. Some rules related to relationship among mental factors are listed (List 1) below.

- **Rule 1:** All universal mental factors occur in all states of consciousness.
- **Rule 2:** Dullness, lack of moral shame, lack of moral fear and restlessness should occur in each unwholesome states of consciousness.
- **Rule 3:** All mental factors either exist in wholesome or unwholesome states of consciousness
- **Rule 4**: All beautiful mental factors occur in the wholesome states of consciousness.
- **Rule 5:** All the unwholesome mental factors occur in the unwholesome states of consciousness.
- **Rule 6:** A mental factor Dullness causes the other mental factors: Greed, Error, Hate, Special doubt and Restlessness also to be arisen in the state of consciousness.
- **Rule 7:** Sloth and Torpor both occur together in a consciousness.

List 1: Some Rules of Mental Factors

3.3 Relating mental factors with physical world

The Buddhist philosophy does not merely talk about the mental factors, but their relationship with the physical aspects of living beings. There are various ways to associate mental factors with physical aspects. For example, a visual object in the physical world can be sensed by the eye, which belongs to physical body of a human being. When an eye contacts with a visual object, in the first place, one's mind forms a consciousness that comprises seven universal mental factors to begin the sensing of the object. As such some collections of metal factors have a mapping to corresponding physical aspects.

Buddhism has broadly classified mental factors into five categories, namely, doors, feelings, functions, objects and roots. Under each of this category there are particular set of mental factors. This classification is too broad and describes how certain groups of mental factors map to physical aspects and some specific types of consciousness. Here doors and objects are physical aspects, while feelings and roots are mental aspects. Each consciousness also has a specific function. In order to expand the Mental Ontology to be more comprehensive, it is appropriate to introduce the Ontology contacting the above five categories. This inclusion enables demonstration of relationship with not only physical aspects, but also the some extra features of consciousness.

We postulate that mental factors can be used as the basis for modeling of mental (consciousness) and physical phenomena in the world.

4. On computing Mental Ontology

In order to model the Mental Ontology, we identify each mental factor as an Object having specialized attributes and behaviors.

4.1 How to model a Mental Factor

A mental factor is attributed by different characteristics, such as object type, category type and general states etc. List 2 provides comprehensive details about some of the identified attributes the mental factor object. A mental factor can arise or take away its position from the state of consciousness and also perform further actions like change the form of appearance and change the associate weight etc.

- **Object type:** Basically the class it belongs to,
- **Group type:** Either it is universal, particular, unwholesome or wholesome
- **General states:** Either it is optional or mandatory to be occurred with other mental factors
- **States in the group:** whether it is mandatory or optional within the group, for example, mental factor Greedy should occur with other unwholesome mental factors but not with all mental factors.
- **Is Dominant:** To specify whether mental attribute is leading attribute. For example Greedy is leading attribute in unwholesome consciousness.
- **Is Dominant with Constraint:** To specify whether mental attribute is a leading attribute if some conditions are satisfied.
- **Allowable Forms:** A mental factor may appear in different format. For example, Greedy can appear in different forms such as desire, attachment, lust, love, gluttonous etc.
- **Forms Appear Constraint to Others:** A mental factor can take different form based on other factors and their weights.
- **Is Subject to the Doors:** Specify whether mental factor is subjected to a door like ear, eye, body, etc.
- **Subjected Doors:** If mental factor is subjected to the doors, then specify.
- **Disjoints Mental Factors:** Specify disjoint mental factors.
- **Associate Mental Factors:** Specify associate mental factors.
- **Constraints on Associated Mental Factors:** Whether associations are bound to specific constraints. Need to specify the weight of each mental factor also.
- **Directly Opposite Mental Factors:** Specify whether a given mental factor has completely opposite mental factors. For an example Greedy has an opposite mental factor called Disinterestedness.

- **Active Status:** Specify whether mental factor is active at the time of decision making.

List 2: Some Attributes of a Mental Factor

List 3 shows identified behaviors of mental factors. These behaviors are important to model the change of mental factors under different circumstances.

- **Occurs:** A mental factor can initially occur with others.
- **Destroy:** A mental factor can take away its position.
- **Change Appearance:** A mental factor can change its form. For example, Greedy can appear in different forms such as desire, attachment, lust, love, gluttonous etc.
- **Change Associate Weight:** A mental factor can change its associate weight with other mental factors.
- **Prompted Action:** A mental factor can cause different actions to be arisen. For example, Greedy can cause to thief, mislead, and lie.
- **Caused Action:** A mental factor may be arisen because of actions occurred in the environment.

List 3: Some Behaviors of a Mental Factor

The class diagram of the mental ontology is shown in Figure 2. The Mental Factor is the base class of the system and each specific mental factor can extend the base class.

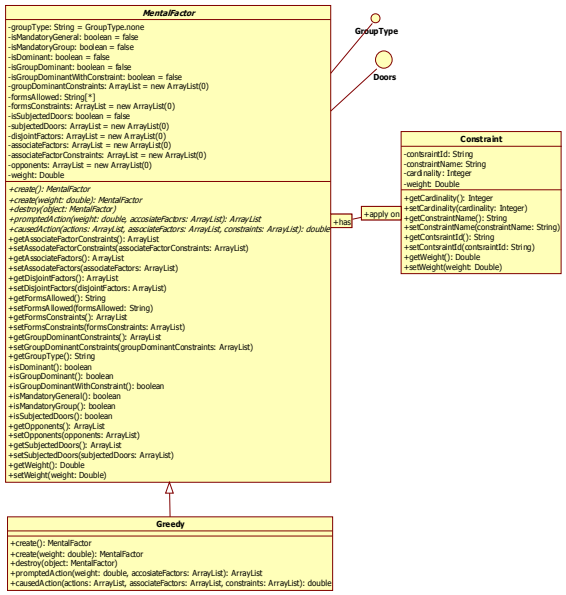


Figure 2: Class Diagram of the Mental Ontology

Following Java code segment shows the implementation of mental factors. List 4 shows the code segment of the base class for the all mental factors.

```
package mentalFactor;

import java.util.ArrayList;

import references.GroupType;

public abstract class MentalFactor{

/* Object belongs category*/
protected String groupType = GroupType.none;

/* Object is mandatory to be occured with other mental facotors*/
protected boolean isMandatoryGeneral= false;

/* States in the group - mandatory within the group*/
protected boolean isMandatoryGroup = false;

/*Specify whether Dominant in the group */
protected boolean isGroupDominant = false;

/*Specify whether Dominant in the group with some constraint*/
```

```
protected boolean isGroupDominantWithConstraint = false;

/*Constraints on Group Dominant - ArrayList accepts the Object of type Constraint*/
protected ArrayList groupDominantConstraints;

……
……...

/*Forms Appear constrained to the others - ArrayList accepts the Object of Type Constraint*/
protected ArrayList formsConstraints;

/*Specfy mental factor is Subjected to the Doors*/
protected boolean isSubjectedDoors = false;

/* If any door the mental factor is subjected*/
protected ArrayList subjectedDoors;

/* Disjoint Mental Factors - ArrayList accept the Objects of type Mental Factor*/
protected ArrayList disjointFactors;
/*Associate Factors - ArrayList accept the Objects of type Mental Factor */
protected ArrayList associateFactors;

/*Associate Factors with Constrained- ArrayList accept the Objects of type AssociateFactorConstraint */
protected ArrayList associateFactorConstraints;

 /*Weight Occur in*/
protected Double weight;

/**
   * Method will create the mental Factor
   * @param weight
   * @return
   */
public abstract MentalFactor create(double weight);

/**
 * Method will destroy the Menetal Factor
 * @param object
 */
public abstract void destroy(MentalFactor object);

/**
  * @param weight specifies the weight associate with
  * @param accosiateFactors in the event
  * @return list of prompted actions
  */
public abstract ArrayList promptedAction(double weight, ArrayList accosiateFactors);

/**
  * @param actions specify the action caused
  *@param associateFactors define associate factors
 * @param constraints define the constraint on it
 * @return proposed weight of the mental factor
 */
public abstract double causedAction(ArrayList actions, ArrayList associateFactors, ArrayList constraints);

}
```

List 4: Code Segment of Base Class

The above base class of mental factors can be extended to create any specific mental factor. List 5 shows how the base class of mental factors been extended to create Greedy class.

```
package mentalFactor;

public class Greedy extends MentalFactor{

/** Method will create mental factor greedy
   */
public MentalFactor create() {
        new Greedy();
}

/** Method will create mental factor greedy
  */with weight
public MentalFactor create(double weight) {
         new Greedy(weight);
}

super.groupType = GroupType.UnWholesome;
super.isMandatoryGeneral = false;
super.isGroupDominant = true;
……..
super.associateFactors = new ArrayList(0);
super.associateFactors[0].add(Dullness);
super.associateFactors[1].add(Envy);
super.associateFactors[2].add(LackOfMoral);
super.associateFactors[3].add(Error);
……
```

```
super.disjointFactors = new ArrayList(0);
super.disjointFactors(0).add(Disinterestedness);

}
```

List 5: Code Segment of Greedy Class

As shown in Figure 3, the Greedy Class groupType is set to the Unwholesome to specify Greedy belongs to the unwholesome category, and associated mental factors related to the Greedy have been assigned to the associateFactors list. As such all the mental factors and their association can be implemented.

Figure 3, shows screenshot of the implemented mental ontology development environment.

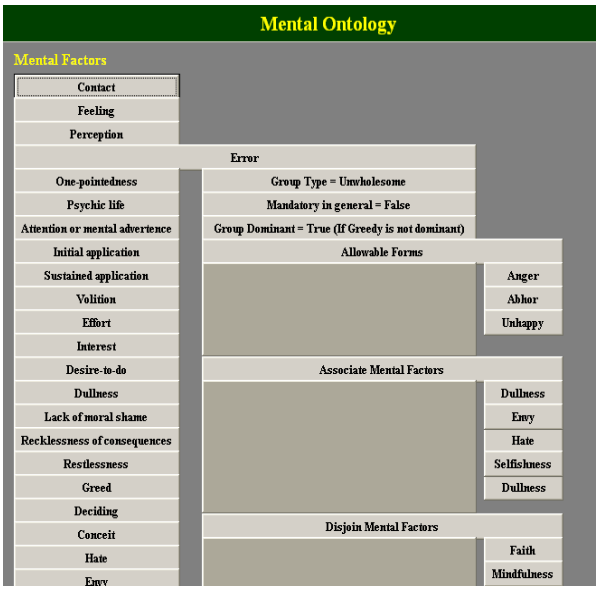


Figure 3: Mental Ontology

4.2 How Mental Ontology works

Here we briefly explain how our Mental Ontology environment can be used model real world systems involving mental conditions such as emotions. Consider agent based intelligent e-commerce system that executes electronic trading between supplier and consumers. An individual agent has been assigned to represent each demand made by consumer and each supplier. The task of demand agent (DA) is to find the best possible matching and the task of supplier agent (SA) is to supply products with maximum profit. Supplier agents are also keen on current demand and other competitive products in the sale. Consider a situation where demand agent (DA1) identifies two competitive products from two different supplier agents (SA1, SA2). This interaction will definitely cause hidden competition between SA1 and SA2 and both of them may want to throw out other from its way. This is a common scenario for all Multi Agent Systems developers. Let us discuss how our Mental Ontology can be used for modeling agents in such a scenario.

Assume that a Multi Agent System (MAS) developer wants to develop his agent, SA1, to work with more powerful agents, say SA2, doing the same task. As such, the developer wishes to consider mental states of SA1. For example, SA1 may be afraid of SA2. Now the MAS developer can use the mental ontology and identify the fear as a form of Error and also associated with mental factors including anger, abhor, unhappy, envy, selfishness, dullness and greedy. Thus the MAS developer can address the fear in terms another mental factor such as abhor, which encourages SA1 to be detached from SA2. In other words, SA1 will be restricted to interaction with SA2. Alternatively, SA1 can go ahead with envy and cooperate with another agent who is also a competitor for SA2. In this manner, MAS developer can introduce various strategies to own agent so as they can deal with agents in competitive environments. Of course, the agent that works in cooperate environments can also be modeled with mental factors pertaining to kindness, sympathy, happiness, etc. Furthermore, less active agents can be motivated through the interaction with the agents who have desire to do things.

The key point in our research is that a mental factor can be used as means of modeling the performance of an agent, and handled on the basis of associated mental factors, instead of direct manipulation of the mental factor. Since people are generally unaware of associated mental factors, the above process could not be captured. Our mental ontology provides a solution for extensive manipulation of mental factors.

5. Discussion

Our literature survey identified the necessity of implementing mental aspects of Ontology as a solution to the current issues in Ontological

modeling. Therefore, we exploited the Buddhist philosophy and postulated that the mental factors can be used to build the Mental Ontology. In our research, we defined mental factors as a class with various attributes and behaviors. A base class has been designed and implemented with the use of Java. It was shown that the base class can be extended to obtain any mental factor. Further, the base class can also be modified to incorporate more attributes and the behaviors that we will identify in our ongoing research.

Further work of this project includes development of a Mental Ontology plug-in for the existing environments for Ontological Modeling.

6. References

[1]. Razmerita, L., Angehrn, A., Maedche, A., (2003), Ontology based User Modeling for Knowledge Management Systems, Available at: <http://www. calt.insead.edu/ Project/ OntoLogging/documents/2003-UM- Ontology_ based_ user_ modeling _for_Knowledge_ Management _Systems.pdf>

[2]. Jin, Y., Decker, S., and Wiederhold, G., (2003), OntoWebber -Model-Driven Ontology-Based Web Site Management, *in: Proceedings of Twelfth International World Wide Web Conference,* Budapest, Hungary. Available at: <http://www-db.stanford.edu /~yhjin/docs/owedbt.pdf>

[3]. Sridharan, B., Tretiakov, A., Kinshuk, (2004), Advanced Learning Technologies, *in: Proceedings of IEEE International Conference* on Volume, Issue, pp. 663 – 665

[4]. Heinecke, J., and Toumani, F., (2003), *A Natural Language Mediation System for* E-Commerce applications: an ontology-based approach, Workshop on Human Language Technology for the Semantic Web and Web Services, *in: Proceedings of 2nd International Semantic Web Conference,* Sanibel Island, Florida

[5]. Garc´ýa-Rojas, A., Vexo, F., Thalmann, D., Raouzaiou, A., Karpouzis, K., Kollias, S., Moccozet, L., and Magnenat-Thalmann, N., Emotional Face Expression Profiles Supported by Virtual Human Ontology, Available at < http://www. image.ece.ntua.gr/papers/449.pdf>

[6]. Karunananda, A.S., and Madurapperuma, A.P., (2006), Towards an Ontology for Kansei. *in: Proceedings of First International Conference on Kansei Engineering & Intelligent Systems,* Aizu-Wakamatsu, Japan.

[7]. Rzevski, G., Himoff, J., Skobelev, P., (2006), Magenta Technology Multi-Agent Logistics i-Scheduler for Road Transportation, *in: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, ISBN:1-59593-303-4, pp 1514 - 1521

[8]. Rzevski, G., (1997), A Framework for Designing Intelligent Manufacturing Systems, *Computers in Insudtry*, Elsevier Science Publishers B. V.,pp 211-219.

[9]. OntoWeb, 2002. Ontology-based information exchange for knowledge management and electronic commerce, IST-2000-29243.

[10]. Burstein, M.H. (2002), Ontology Mapping for Dynamic Service Invocation on the Semantic Web, American Association for Artificial Intelligence.

[11]. Karunananda, A.S., Rzevski, G., (2006), Ontological Modeling: State of the Art, Unresolved Issues And New Research Directions, Available at: < http:// rzevski.net/ paperlink.asp?u_link=pdf/Ontological%20Modelling%20Paper%203.10.06.pdf>

[12]. *Shiizuka, H and Watada, J*, (2006) Overview of the Kansei system, *in: Proceedings of the I MECH E Part I Journal of Systems & Control Engineering,* ISSN 0959-6518, pp. 659-665(7).

[13]. Asoka S. Karunananda (2007), An Approach to humanizing e-Learning systems, *in: Proceedings of the 2nd International Conference on Industrial and Information Systems,* IEEE Sri Lanka Section, August 9-11, Peradeniya

[14]. Narada Maha Thera, (1979), A Manual of Abhidhamma,
Buddhist Missionary Society, Malaysia.

[15]. Gruber, T.A., (1993), A Translation Approach to portable Ontology Specification, Knowledge Acquisition, 5(2) 199-220.

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Using an Intelligent Agent with PBL approach to Online tutoring and mentoring

U. Weerasoooriya & A. Madurapperuma

Faculty of Information Technology, University of Moratuwa, Sri Lanka.

E-mail: udayanthi@itfac.mrt.ac.lk , ajith@itfac.mrt.ac.lk

Abstract

Online Tutoring and Mentoring is an essential activity for effective e-Learning systems. However, at present Online tutoring/ mentoring is conducted merely on the basis of personal experiences of tutors and mentors, without a theoretically-based approach. This paper presents an approach that exploits the theory of Problem-Based Learning (PBL) for online tutoring and mentoring. The proposed approach drives the online tutoring/mentoring process through a four dimensional framework concerning; identification of facts, generating ideas, identifying learning issues and preparing an action plan. In our approach, online tutor/mentor is simulated as Agent software that runs on a Learning Management System of an e-Learning environment.

1. Introduction

Nowadays, e-Learning has become popular than ever before due to its ability to provide more flexible learning strategies to suit with individuals. In this sense, e-Learning has been used in many countries all over the world to deliver short courses, undergraduate programmes and Postgraduate programmes. For example, Indira Gandhi National Open University of India and Open University of UK offer most of their Undergraduate and Postgraduate programmes through e-Learning [1]. Even traditional universities are moving towards e-Learning by offering some of their programmes online. Another example is the tutorials available via e-learning. In e-Learning, the online learners will have the advantage of learning at their own pace, own time at their own place and the flexibility they can have in their learning process [1]. Despite the advancement in technologies to offer e-Learning strategies, the value of the teachers (or tutors) involvement in learning process is inevitable. Researchers have shown the value of humanizing e-Learning systems [2]. Since e-Learning does not encourage the physical presence of a teacher, the role of tutor/mentor as the facilitator of the learning process becomes an essential element of the process.

In fact the role of tutor/mentor is more important for e-learning than for face-to-face learning [2]. This is because; online learners tend to feel that they are kept alone and no opportunity for them to get resolved their learning issues instantly. As such, online tutor/mentor plays a key role in motivating and getting learners actively involved during a learning session [3]. Obviously, the role of an online tutor is more difficult than the tutor's role in the face-to-face environment. At present, online tutoring is operating as a process conducted merely by the experience of individual online tutor/mentor, and there is no theoretical approach to guide the online tutoring/mentoring process. Thus, there is a need for developing a theoretical model to drive the process of online tutoring/mentoring to ensure that online learners are facilitated during a learning session.

Our study for addressing above issue has revealed that emerging trend in Problem Based Learning (PBL) could be exploited to develop a model for emulating the role of tutor/mentor. PBL has won wide spread recognition as a learning strategy that inculcates better skills to work in the industry and also to pursue higher studies [5, 6, 7]. It is evident from our research that PBL process is similar to what an online tutor/mentor performs during sessions such as moderation of a discussion forum/chat session and motivating isolated learners. Therefore, we have decided to exploit a PBL model concerning four (4) dimensions to drive a learning process. These four (4) dimensions are identified as

identification of facts, generating ideas, identifying learning issues and preparing an action plan.

This paper presents the theoretical model based on the PBL theory to familarize the online tutoring and mentoring.

The rest of the paper is organized as follows: Section 2 provides an overview of e-Learning, emphasising on online tutoring and mentoring. Section 3 describes the PBL process and identifies a model that can be used in e-Learning. Section 4 discusses the novel approach to facilitate online tutoring and mentoring. Section 5 describes the designing of Agent software for online tutoring and mentoring. Section 6 concludes the paper with a discussion.

2. Overview: E-Learning

e-Learning refers to the use of Information and Communications Technology (ICT) to support learning in general. This ranges from the use of educational software running on a stand alone PC to online learning materials made available on the World Wide Web. It is widely accepted that Web technology and multimedia technologies have revolutionized e-Learning. This is because, Web technology provides not only a multimedia-based presentation environment for e-Learning but also effective communication means with the use of e-mail, discussion forums, chat sessions, etc.

Therefore, the teacher's role will be played by the e-learning materials and an online tutor/mentor will be there to facilitate the learner. A learning process to happen through e-Learning mode or through face-to-face mode, there should be a tutor to facilitate the learner throughout the learning process [8]. One can argue that the tutor's role is not needed in an online learning environment since the learning materials are expected to provide the necessary guidelines for self-learning. However, to reduce the learner isolation and to increase the communication there should be an online tutor/mentor who ensures the active participation of the learner in the different learning activities [9].

As compared with traditional face-to-face learning, e-Learning has three (3) major advantages. Firstly, learner has the ability of learning in his/her own phase. Hence, the learner can use his own free time to study. The second advantage is that the learner can study on his/her own place and do not have to be at a specified place. Thirdly, e-Learning provides the learners with more flexible learning strategy, within which learners can use their own learning styles.

Nevertheless, there are some disadvantages too, which might occur in e-Learning. Since the learner is learning at a remote place, she/he might experience technophobia and might face difficulties in fixing the technology problems on his/her own. In e-Learning, the learning materials are available in the electronic format and some of the learners will find it difficult to learn using these compared to printed materials. Another major problem is the limited interaction happening among the learners. In the traditional face-to-face learning, students have an opportunity to directly interact with peers and teachers. Figure 1 illustrates the interaction in face-to-face learning. In face-to-face learning, direct interacting between teacher and the learner enables resolving of learning issues instantly.

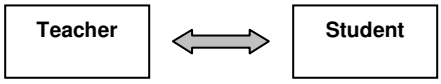


Fig 1: Interaction in Face-to-Face Learning

Furthermore, in face-to-learning, students will have the chance of meeting their friends and discuss the learning issues and share and construct the knowledge in broader manner.

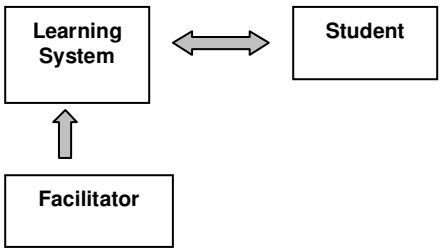


Fig 2: Interaction in e-Learning

In contrast, in an e-Learning mode, learners will directly interact with the online learning system without a direct support from a facilitator. Figure 2 illustrates typical model for interaction in an e-learning environment.

Online tutors and mentors are considered as the facilitators in the model shown in Figure 2. Below is a brief description about the role of online tutors and mentors.

A. Online tutoring/mentoring

Online Tutoring and Mentoring is inevitable for effective e-Learning systems. Unlike a student learning in face-to-face learning mode, an e-Learner will sometimes experience isolation, due to absence of the teacher and peers in the e-Learning environment. Therefore, the online tutor/mentor should be able to facilitate the learners by reducing the isolation and maintaining better interactions with them. Online tutor/mentor should be able to provide relevant guidance in activities such as quizzes, discussion forums, chat sessions and assignments, so the learners will be able perform these activities successfully.

Online tutors/mentors should have a set of special characteristics to perform their duties well. They should be able to motivate the learners throughout the whole learning process by giving positive feedback on time without any delays. The online tutor/mentor needs to write to the learner individually encouraging the learner participation in the specific activities. They should encourage the collaborative learning activities, thus helping the learners to reduce the isolation from the social community [10].

In general, online tutoring and mentoring skills are to be gained through the knowledge of the past experiences, which results in the process becoming personal to the individual tutor/mentor. As such, the current practices to online tutoring have no theoretical-basis, and have a negative effect on the uniformity and quality of services of online tutoring and mentoring. Next section discusses how we exploit theory from Problem-Based learning towards the development of theoretical model to online tutoring and mentoring.

3. Problem Based Learning

A. Importance of PBL

Problem Based Learning (PBL) refers to an instruction method in which the teacher builds his instruction around a practical problem and encourages students to discuss the problem in small groups, with an aim to cultivate active learning, critical thinking, and problem-solving skills among them [6]. PBL facilitates learners to learn to learn. This approach inculcates the learners with many skills including ability to identify a problem, work in teams to solve problems and a variety of soft skills required to solve real world problems [11]. Explicitly PBL is promoted as encouraging the integration of theory and practice, enhancing lifelong learning skills and developing professional competence [3]. It is suggested that institutions that adopt PBL do so because of its potential to deliver a competitive advantage to the institution as much as for educational reasons [6]. It is argued that PBL is being really beneficial for the students to be successful in their learning process [1]. Evidences has shown that PBL is an effective tool to foster students developing the critical and creative thinking skills and enhance their innovation capabilities through the process of problem solving [1].

B. Characteristics of PBL

Problem Based Learning facilitates the learners to learn the subject matter by solving practical, real world problems. Unlike in most other learning methods, PBL covers the breadth of the subject matter rather than the depth of the subject matter. [6]. There are three (3) essential characteristics of Problem Based Learning. The PBL process focuses on problems which are open ended and challenging [5, 6]. In PBL process, learners work in small collaborative teams to solve the given practical real-world problem. The teachers play a facilitator's role instead of the teacher's by directing the students to follow the PBL process.

Before the Problem Based Learning Process begins, it is essential to form teams consisting around five (5) learners in one team. Though, it is possible to select the team members randomly, the most suitable way is to formulate teams with a help of a Quiz which helps to identify the different team roles. The facilitator arranges the teams considering the different team roles which the different learners can play, consequently each team consist of team members that form a good team which consists of all different team roles.

Each team has a facilitator who helps the learners in their Problem Solving Process. The team members will have to conduct meetings time to time with their facilitators. These team meetings will focus on discussing the new things the learners have learned and to clarify issues with the help of the facilitator, if there are any [11]. In order to carry out the PBL process successfully, the team members meet at least once a week. The total time required for solving the problem will depend on the size of the problem scenario and as an average it takes about three (3) consecutive months. It is a required to go

through the PBL process and gain the necessary skills in facilitating, before facilitating the learners [5].

C. How PBL process works

The PBL process comprises of the following steps in constructing a solution through team work for a given problem [11, 12].

- Identification of facts from a given problem stated
- Generating the ideas based on the facts given
- Generating the learning issues
- Generating the action Plan

The procedure in applying the above steps in a PBL process works in the following manner. At the first meeting of the PBL process, the students meet their facilitator and agree upon the plans of communication among the team members and the facilitator. This can be via email or through regular meetings where all the members will be present on any predefined date, on a predefined time and in a specific place. Moreover, it is possible to have the meetings via synchronous chat sessions or online discussion forums. During the next step, the team members identify the different roles each plays within the team. For example, team leader to lead the team, a secretary to record the minutes of their meetings, etc… The real PBL process begins soon after this. Next meeting date should be decided before winding up a team meeting. The facilitator distributes the real-world problem scenario to the learners and the learners are required to go through the problem and identify the facts given in it. The facts are the information which directly extracted from the Problem without making any assumption. After identifying all the possible facts from the problem, the individuals in the team generate ideas leading to a solution. Since the Problem scenarios given to the learners are open ended ones, there can be more than one solution for any given problem. Therefore, the learners come up with several different solutions to the problem. The next step in the PBL process is to identify the learning issues. Learning issues are the areas that the team needs to do research on, in order to get the knowledge in solving the problem. After the first meeting, the learners will have to search for relevant information in library books, magazines or whatever the printed materials of relevant resources. They will have to find additional information from the various different resources available on the World Wide Web. The final step of the PBL process is to identify the action plan. All the activities that have to be carried out by the different team members should come under this category. For example, if the team needs to interview somebody to get some additional information on certain area, the action plan should contain an entry about the interview and the person who is responsible for that task.

Soon after the first meeting, the team will have to collect the necessary information needed to reach the solution individually. At the meetings the team members are required to present their individual findings and to share the information with others. The facilitator will play a major role her e, by observing whether the team is doing the right thing and directing the team towards the correct path. Depending on the information presented by the different team members and the results of the team discussions, the Learning issues and the Action plan will have to be revised accordingly sometimes at every meeting.

4.Approach:Online Tutoring/Mentoring Agent

In our approach to theoretical-based solution for Online tutoring and mentoring, we exploit the four steps of the PBL process to develop a theoretical model for online tutoring and mentoring and develop agent software using this model.

We propose to implement the PBL four-step model as a software Agent that can be delivered as a plug-in for a standard Learning Management System (LMS) such as WebCT [14], Blackboard [15] and Moodle [16]. LMS would be the best e-Learning environment that can be expanded with a special plug-in to implement the role of an online tutor mentor. This is because, by default, a typical LMS provides most communication facilities such as e-mail, discussion forums and chat sessions that are required by an online tutor/mentor.

Development of our online tutor/mentor Agent is done on the Moodle, a well known and one of the most successful LMS todate. In addition, being a Free and Open source software, it is possible to do development work on the Moodle to integrate with other software.

In addition to standard support for discussion forum, chat sessions, etc., Moodle provides with connection to Wiki like document management systems that are useful in PBL process. Moodle also provides with the facility of a Grade Book

which indicates the marks for different activities out of specified marks, and the time taken to complete an activity out of specified time. By regularly monitoring the Grade book, we can track the progress of the learner easily. For example, we can see whether the learner has completed an activity in the specified time duration and also whether his/her grade is satisfactory. Also it is possible to compare the grades of the different learners. In this particular scenario, if the time taken to complete the activity is greater than specified time or if the grades are not satisfactory then it is obvious that the learner is not up to the standards. Therefore, he/she needs help to improve him/her self. Since he/she learns at a distance, our Agent software has to play a major role here by directing him towards the right path to achieve the learning objectives.

As the first step of the PBL approach, our online tutor/mentor Agent will identify the activity that the learner has failed to do. For this purpose, it uses the Grade book facility of Moodle to retrieve

- the marks for different activities out of specified marks and
- the time taken to complete the activities out of specified time duration

As the second step, it will find out the reasons for the problem. Therefore, the online tutor/mentor analyses data and get the results. For example, if the time taken to complete the activity is greater than the specified time, then the learner has a problem with managing his/her time.

As the third step, the online tutor/mentor Agent will identify the areas in which it could provide guidance for the learner. If we consider the same example mentioned above, then the online tutor/mentor has to provide with the help to the learner to manage his/her time in doing activities. So it will offer the learner with a similar activity that has been covered during the session. These activities are taken from a library which consists of different type of problems that can be solved using Problem Based Learning. Since the Agent has the ability of monitoring the performances of all the learners, it will identify the similar type of learners and help them to form teams that they can work together.

As the final step, the online tutor/mentor Agent monitors the learner closely while he/she engages in the activity, and provide with the necessary guidance when the learner needs it. Since the learner tries to solve the activity using PBL, the online tutor/mentor agent monitors the times taken to go through the each activity and if the learner takes more time, the Agent helps the learner by indicating the time remaining to complete the activity.

The facilitation provided by the online tutor/mentor Agent will differ according to different learners. It can serve all the learners who connect with the Learning Management System via their individual user profiles. Depending on the grades that the Grade book specifies, this different learners will receive different guidance and the problems to solve by our online tutor/mentor Agent.

It can also be justified the choice of Agent technology [17, 18] for implementing the PBL process into online tutoring and mentoring due to the following reasons. By definition, Agent is a kind of automated software that can support the user without requiring repeated intervention and request from the user. In addition, Agents can learn from past experiences and evolve to provide better solutions in subsequent events. Furthermore, an Agent can handle several users at the same time through the profile of the users. On the other hand, there are so many open source Agent software available, and they can be readily used to develop our online tutor/mentor Agent without requiring to code an Agent from the scratch.

5. Design of Online Tutor/Mentor Agent

Online Tutor/Mentor Agent has been designed to work as a plug-in to standard LMS as shown in Figure 3. This Figure gives an overview showing the entire solution comprises three major components, namely, client learners, LMS running on a server and Agent as a plug-in to LMS.

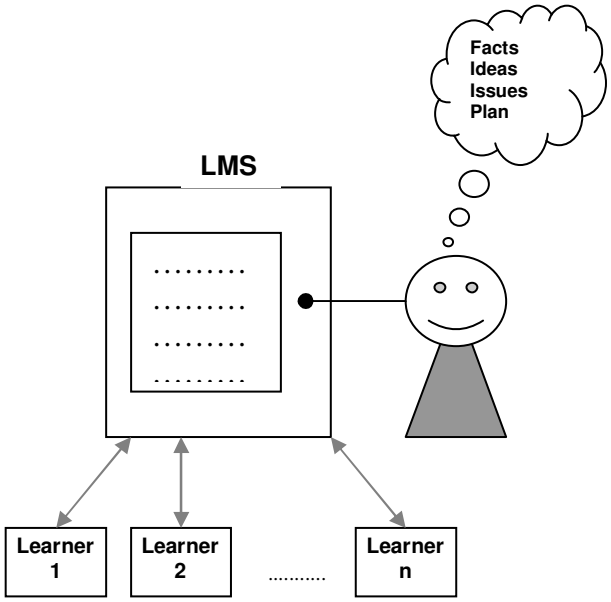


Fig 3: Online tutor/mentor as a plug-in to LMS

A. Learners

A learner uses a standard web browser for interacting with LMS. There is no special configuration or setting from learners' side. The learners do not even need to know that there is an Agent running on the system. Since the Agent automatically comes in action when necessary. If the performance of the learner is not up to the standard, the online tutor/mentor Agent automatically comes in to action and provides him/her with the necessary support to achieve the learning objectives. If the learner needs any support apart from what he is automatically getting from the online tutor/mentor Agent, then the learner has to request it from the Agent. This request can be done via email, or using short message service available in Moodle. It is also possible to make a request through a chat session or through a discussion forum created for the communication purposes with the online tutor/mentor Agent. The different learners receive different support from the online tutor/mentor Agent according to their performances. Therefore, the learners feel that they are getting a unique service for him/her from the online tutor/mentor agent but not a service that is common to everybody who engages learning in the Learner Management System.

B. LMS

This is a standard Learning Management System. It will provide with the access to different learning resources like files located in remote places or other different multimedia web resources. The LMS also conducts chat sessions which allow synchronous communication among learners. LMS allows another type of asynchronous communication via discussion forums. LMS has another type of activities called quizzes which helps the learners to assess their performance by them selves. Assignments are another type of activity that included in the LMS. To manage the documents and to get help with the collaborative online activities Wiki like document management modules are available with the Learning Management Systems. Like the above mentioned activities, it is possible that our Agent is plug-in into the LMS in such a manner that LMS administrator can activate the Agent if necessary.

C. Tutor/Mentor Agent

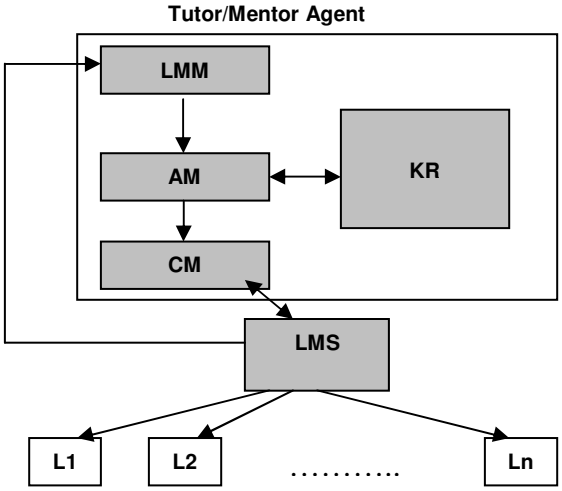


Fig 4: Architecture of the online tutor/mentor

LMM – Learner Monitoring Module
AM – Analyser Module
CM – Controller Module
KR - Knowledge Repository
LMS - Learning Management System

Figure 4 illustrates the Architecture of the online tutor/mentor Agent. This is the core of this research work. Our intelligent tutor/mentor Agent has the capability of handling several learners at a given

time. Here the Agent has been designed to instantiate a leaner profile for each learner and maintain the profile. The intelligent Agent creates an instance when the learner logs in each time. During the process of learning the Agent monitors whether the learner has actively participated in the prescribed activities in the learning session. This can be done with the use of Learner Monitoring Module. The online tutor/mentor Agent then analyses the information received from the LMM by the Analyser Module and if learner appears to be isolated, performs below other learners, or performs below the specified standards, etc., the Agent assists the learner to be active as per the steps in the exploited PBL model. Since the Agent can have access to performance of a learner, it can provide additional learning advice. The analysed information receiving from the Analyser Module will be stored in the Knowledge Repository. When it is necessary to provide additional learning advices, they are provided with the help of Controller Module. These advices can be provided through email, short message service available on Learning Management System, predefined chat sessions that enables synchronous communication or via discussion forums which enables asynchronous communication. Online tutor/mentor Agent also provide with the help by introducing simple PBL sessions to be completed with the similar type of learners. It also helps the learners by identifying suitable colleagues to form teams. For this purpose, the Agent can maintain a library of problems which are suitable for Problem Based Learning related to topics covered in a session.

6. Discussion and Further Work

We have presented a novel approach to effective online tutoring and mentoring. This approach has several advantages as follows. In the first place, it provides a theoretical-basis for online tutoring and mentoring. Since the solution has been presented as an Agent running on an LMS, many learners can be accommodated at the same time. This is a very big advantage as a typical online tutor can handler only few students at a time. Further, the Agent can provide uniform support for all learners, and this may not be the case with human online tutor/mentors if they are not equally competent.

It is possible to activate the Online Tutor/Mentor Agent while human online tutors are involved in tutoring and mentoring. This process allows the Agent to learn from good habits of human online tutors/mentors, and improve the Agent performance gradually beyond the human tutor/mentor. Therefore, it is possible to use the online tutor/mentor Agent as an assistant to online tutors/mentors. In the long run this will enable to cut down cost incurred in hiring experienced human tutors/mentors for e-Learning processes.

The online tutor/mentor Agent can perform some tasks, which are in fact cannot be done by a human tutor mentor. For example, learners competency to use LMS and associated tools can be instantly monitored by the Agent, and provide advise to improve. Further, online tutoring by human is generally operates in asynchronous manner, and this does not allow the learner to get the tutoring/mentoring support in a timely manner. However, our Agent can operate all the time, notice the requirement of help and promptly attend the matter.

The Agent can also be extended to enable communication with the LMS through mobile devices. The current Moodle like LMS does not enable such communication, except through PC clients.

In summary the following are the key benefits

- Looking after many students simultaneously
- Improve beyond human online tutors
- Operate timely manner
- Ability to work as an assistant for human online tutors/mentors
- Reduce the cost included in hiring human tutors/mentors
- Automated communication with LMS

Features of the Agent have already been identified and the design model been prepared. Implementation of the Agent commence soon on the Moodle environment.

7. References

[1] Panda S., "Planning and management in distance education", Open and Distance Learning Series, pp. 195-205, 2003

[2] Karunananda A. S., "Theoretical-based approach to e-Learning" In Proc. Proceedings of the International Conference on Industrial and Information Systems, IEEE Sri Lanka Section, August 8-11, 2006

[3] Greening T., "Scaffolding for Success in Problem Based Learning", Medical Education Online [serial online], vol. 3(3), 1998.

[4] Lee T., Chang C., and Dong G., "Building an Interactive Caring Agent for Students in Computer-based Learning Environments", Seventh IEEE International Conference on Advanced Learning Technologies, 2007

[5] Barrows, H. S., "A taxonomy of problem-based learning methods", Medical Education, 20, 481-486, 1986.

[6] Hong J., "Strategies for constructing Problem-Based Learning Curriculum", University of Tampere, 2003

[7] Boud, D. & Feletti, G., "The Challenge of Problem Based Learning", New York: St. Martin's Press. 1991.

[8] Savery J. R. and Duffy T. M., "Problem Based Learning: An Instructional Model and Its Constructivist Framework", Educational Technology, vol. 35(5), pp. 31-38, 1995.

[9] Karunananda A. S., "Learner's Perspective driven ontology for e-learning", In Proc. Eight International Conference on Humans and Computers, Japan, pp 143-149, 2005.

[10] Gunawardena, C. "Designing the social environment for online learning: the role of social presence", Distance Education and Technology: Issues and Practice, pp. 255-270, Hong Kong: Open University of Hong Kong Press. 2004.

[11]Conway J., Little P., Parker V., "Cultural Congruence or Cultural Conflict: Challenges in implementing Problem Based Learning across cultures", unpublished

[12]Peterson M., "Skills to enhance Problem-Based Learning", Medical Education Online [serial online], vol. 2(3), 1997

[13]Johnson W.L., and Shaw E., "Using agents to overcome deficiencies in web-based courseware", In Proc. AI-ED Workshop on Pedagogical Agents, pp. 48-55, Japan, 1997

[14]WebCT, http://www.webct.com/

[15]Blackboard, http://www.blackboard.com/

[16]Moodle, http://moodle.org/

[17]Rickel J., Johnson W., "Animated pedagogical agents for team training", In Proc. ITS Workshop on Pedagogical Agents, pp. 75-77. 1998.

[18]Shaw, E., Johnson, W.L., & Ganeshan, R. "Pedagogical agents on the Web", In Proc. Third Annual Conference on Autonomous Agents, ACM Press, pp. 283-290, 1999

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

NLP - Based Expert System for Database Design and Development

U. Leelarathna [1], G. Ranasinghe [1], N. Wimalasena [1], D. Weerasinghe[1], A. Karunananda [2]

Faculty of Information Technology, University of Moratuwa, Sri Lanka.

cinco.publish@gmail.com[1] , asoka@itfac.mrt.ac.lk[2]

Abstract

Database designing and development involves a sequence of tasks including extracting the requirements, identifying the entities, their attributes, relationships between the entities, constraints, drawing a conceptual schema, mapping the ER diagrams to the database schema, and eventually developing the database. As such database design and development has become a tedious task for novice person. In addressing the above issue, we propose a Natural Language Processing enabled Expert Systems, which accepts textual domain descriptions and generate a relational database schema followed by a database. The entire NLP enabled system can be customized to link up as a front-end for any database management system. The system has been developed using Prolog, Flex and C#.net.

1. Introduction

Database and Database systems have become an essential component in modern day life. Most of the applications that we come across today involve large amount of data, making database systems the first choice in storing, processing, and disseminating data.

Designing a database requires a good knowledge and skill on database concepts. After a thorough requirement analysis, the conceptual schema design should be done in a high level data model such as Entity Relationship Model. Then this should be mapped in to a schema using a data model such as Relational Data Model.

Hence the user must have some idea of ER designing and mapping it to Relational schema in order to design and implement a database for an application. This makes designing and implementing a database a difficult task and complicated process for a user who has less knowledge on databases. The expert knowledge and skills is always preferred in this context to develop a superior database. However, consulting and depending on expert knowledge all the time, brings out certain issues, since expert knowledge is expensive and not always available.
If there's a possibility that users can represent the requirements in their natural language rather than as ER diagrams or Relational schema and the system can extract the user requirements itself, draw ER diagrams and implement the database automatically, then life will be easier for the user. Even a novice user with less knowledge on database design will be able to develop a database easily using such a system.

Based on this idea we have been working on developing an intelligent system to design and develop a database using expert knowledge with minimum user interaction. The proposed system is a **Natural Language Processing enabled Expert system.** It takes the requirements specified in natural language as the input, process it and identify entities, attributes, relationships and constraints using an expert system, draw the Relational schema and develop the database for the user. This paper presents our approach to develop this system.

The rest of this paper is organized as follows. Section 2 provides an overview of the current approaches to database development. Section 3 describes the design of the system. Section 4 presents how the system works and finally Section 5 presents a discussion with a note on further work.

2. DB Systems-Design & development

Many tools have been introduced to simplify the process of database design and development. Most of these tools facilitate the conversion of ER diagrams to database. These designing tools take the user drawn database designs (e.g. ER diagram), and generates the SQL code appropriate for the database.

Most tools for database design and development facilitate the conversion of ER diagrams to database. These designing tools accept the user drawn database designs (e.g. ER diagram), and generates the SQL code appropriate for the database. For example, Clay [10] is a modeling tool of this type that acts as a plug-

in for Eclipse. DBVA for JBuilder for Windows 2.0 [11], DBVA for IntelliJ IDEA for Windows 2.0 [10] are some of the other similar tools to build database systems.

There are also several tools that use Expert systems for database design and development. Among others, Generalized Expert System for Database Design (GESDD) [3] is one such system. It is made up of two parts:

1. An expert system for generating methodologies for database design, called ESGM

2. An expert system for database design, called ESDD.

Using ESGM, database design experts can specify different design methodologies or modify existing ones. The database designer uses ESDD to design a database. It supports several well-known data models, namely, the hierarchical data model, the network data model, or the relational data model. However, GESDD is a menu-driven system and no support for interaction though natural languages.

SECSI[2] is yet another software system, which uses an expert system for database design. This system generates a specific semantic network representing the application from an application description given with either a subset of the natural language, or a formal language, or a graphical interface. It uses a set of design rules to complete and simplify the semantic network to reach flat normalized relations.

It is understood that although experts systems technology has already been used to develop software tools for design and development of databases, such systems are mainly targeted for experts in database design, but not for novice designers with little experience. Further, although SECS like systems enables NLP supports, such systems do not provide adequate facilities for novice database designers to use the system.

In view of that we propose to design and develop NLP enabled expert system that can be used by a novice database designer.

3. Proposed NLP enabled ES

We have designed the proposed system with 5 modules, namely, NLP Parser, English Morphological Analyzer, Dictionary, Knowledge base and Output generator. Figure 1 shows the top-level design of the Natural Language Processing enabled Expert system**.** In broader sense, NLP enabled expert system comprise two major modules, namely, experts system module and output generator module. The role of each module is described below.

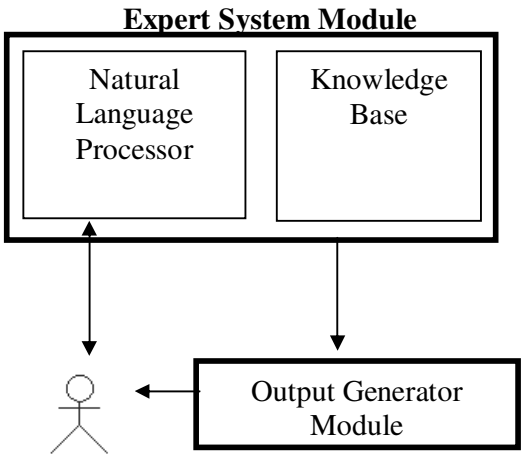


Figure 1: Top Level design of NLP enabled ES

3.1 Expert System Module

The role of the Expert System is two fold:

- Use a Natural Language Processing parser to obtain user information in the best possible way.

- Use the Knowledge Base of expert's knowledge to develop the relational schema of the database.

It is a known fact that the domain descriptions provided by the users are generally incomplete and rather ambiguous. In such a case, with the use of features of the expert system technology, our system is able to handle incomplete information by asking questions from the user. Also after the system design and develop the database, user might want to know reasons or explanations on certain system decisions. In this case system supports reasoning and explaining.

3.1.1 Natural Language Processor

The Natural Language processor in the Expert System Module collectively comprises a tokenizer, a parser, an English Morphological Analyzer and English language dictionary. Figure 2 visualize the Natural Language Processor.

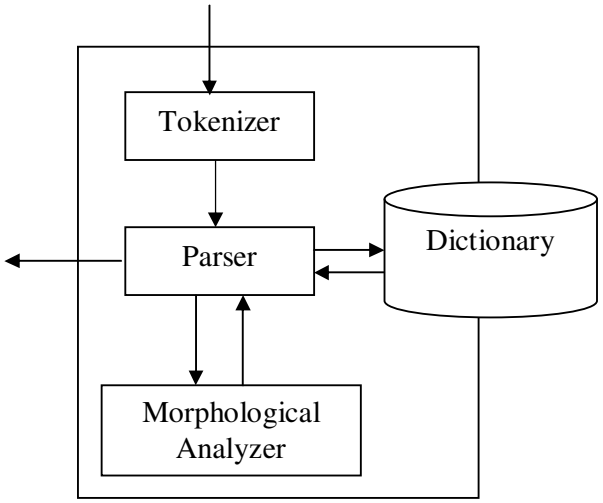


Figure 2: Natural Language Processor

3.1.1.1 Tokenizer

The natural language processor contains an efficient tokenizer written in ISO standard Prolog. Tokenization is the process of breaking a text file up into words and/or other significant units. The tokenizer takes each sentence in the text file one-by-one and breaks them in to separate tokens For example; it breaks the input string "this is an example" into the series of tokens:

[this,is,an,example]

3.1.1.2 English Parser

The English Parser embedded in the expert system module receives tokens for each sentence of the domain description, and identify the parts of speech in the sentence [9].

Each sentence of the file is tokenized and stored into a variable. The value of the variable of a given sentence is passed to the parser. The parser contains the Definite Clause Grammar (DCG) rules [4, 5] that identify the parts of speech. At this stage of the project, we assume that the sentences input to the system, has only present tense and no spelling and grammatical mistakes. The parser in this project is linked up with an English dictionary to get parts of speech [1] information of the words in the input sentence. Win-Prolog is used to develop the parser in Flex environment.

3.1.1.3 English Morphological Analyzer

The natural language processor also deals with an English Morphological Analyzer that identifies the different inflection of the English words [8]. The inflections of each word are attached to the list which is sent to the Expert System. This is also developed using Win-Prolog in Flex environment.

3.1.1.4 Dictionary

NLP system uses a dictionary such as English word dictionary to identify the parts-of-speech of a given tokenized sentence. English word dictionary contains English words and the lexical information. English concept dictionary contains synonyms, anti-synonyms and general knowledge about English words. To identify the parts-of-speech of each tokenized sentences our system uses the WordNet dictionary [6, 7].

WordNet is a lexical reference system, developed by the University of Princeton. Its design makes the use of dictionaries more convenient. Data from WordNet can be used as input for various applications. It provides a database, written in Prolog.

3.1.2 Knowledge Base

The knowledge base consists of the rules that are used to identify database relations and relationships. This is the reservoir of expert knowledge to design and develop databases. It identifies the database entities, relationships, relations and attributes by applying the rules and designs the database relational schema.

The knowledge base has been implemented as a production system that uses rule base knowledge representation. The inference engine of the expert system uses the forward chaining strategy to discover the appropriate design of a database for a given problem. The inference engine also uses the priority base conflict resolution for exploring the knowledge base. The output generated by the expert system is written in a special format to a text file. The Expert System module has been developed using Knowledge Specification Language in Flex environment.

3.2 Output Generator

The output generator reads the output of the expert system and graphically represents the relational database schema to the user. At present output generator provides graphical output separately from the NLP interface. The output generator module also reads relational scheme and generate SQL commands, which will be executed by a respective

database management system. The output generator module can be customized to link up the expert system module with an available database management system. This module has been implemented using C#.NET in Visual Studio.Net framework.

4. How System Works

In this section we describe how the system works for a given input paragraph. Assume that the user has input the sentences **'Department has employees. Employee has id, name, department, age, address'** through NLP interface.

Then the expert system accepts the input stream and sends through the parser and generates the following output.

Sentence 1- [Department/NN, has/VB,
employees/NNS/.]

Sentence 2- [Employee/NN, has/VB, ID/NN,
name/NN, department/NN,
age/NN, address/NN/.]

Sentence 3- [Department/NN,has/VB
ID/NN,name/NN,ID/NN,name/NN,
noofemployees/NN].

The above output has the semantics as
Sentence 1- Department is a noun, has is a verb, employees is a plural noun.

Sentence2 - employee is a noun, has is a verb, ID is a noun, name is a noun, department is a noun, age is a noun, address is a noun.

Sentence 3 -department is a noun, has is a verb, ID is a noun, name is a noun, noofemployees is a noun

The expert system reads the input list and generates the output by identifying the relations, attributes, relationships etc. According to the example the system identifies the first sentence as a description of a relationship and the second sentence as a description of relation. Figure 3 shows the output generated by the expert system which is a text file.

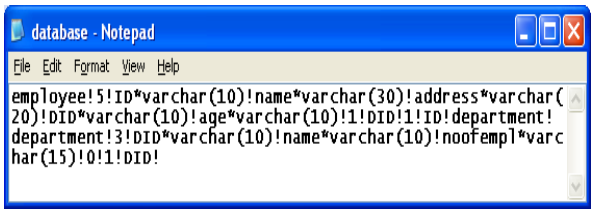


Figure 3: An output generated by the ES

According to the above information the output generator of the module draws the relational schema and develops the database with two tables, namely, department and employee. Figure 4 shows the relational schema generated by the output generator module to indicate the relation via department ID (DID).

.

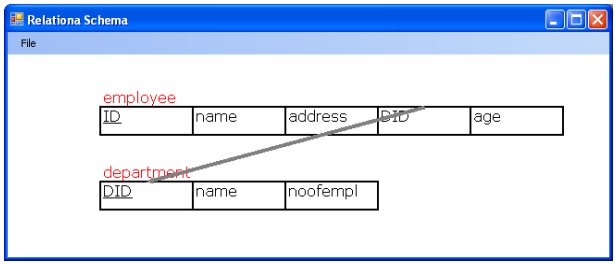


Figure 4: Relational schema by output generator

5. Discussion

The objective of our project was to develop a Natural Language Processing enabled Expert System for database design and development. The proposed system simplifies the database design and development process and unlike many existing systems for database development, it can be used by a novice database designer. The system accepts textual domain descriptions and generates a relational database schema followed by a database. At present the system can handle domain descriptions written in simple present tense. Since the system has been developed as an expert system, it is possible for the system to communicate with the user so as to receive complex versions of descriptions in a simpler form. In addition, since the system is able to provide reasons for the generated designs, the user will be in position to receive justifications for proposed designs. As such NLP technology together with the Expert system technology has made the database design and development process more interactive and more flexible even for the use by novice persons.

This system can be further improved by integrating the output generator and the expert system interface in to a single user interface, so that the user is able to view the relational schema and do modifications

when necessary. We can also provide the user with the facility to customize the output generator module to be linked up with a chosen database management system. Incorporation of more experts knowledge for the knowledge base and additional knowledge for improving the NLP module will improve the effectiveness of the system.

6. References

[1] Aravind K. Joshi, B. Srinivas "Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[2] Bouzeghoub M., Gardarin G., Métais E., "Database design tools: an expert system approach", Proceedings of the 11th international conference on Very Large Data Bases - Volume 11, Stockholm, Sweden, 1985 pp 82 – 95

[3] Dogac A., Yürüten B., Spaccapietra S. "A Generalized Expert System for Database Design", http://portal.acm.org/citation.cfm?id=63389

[4] Doran C., Egedi D., Hockey B. A., Srinivas B., Zaidel M., "XTAG System – A wide coverage grammar for English.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[5] Egedi D, Martin P., "A Freely Available Syntactic Lexicon for English.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[6] Gangemi A., Guarino N., Oltramari A., "Conceptual Analysis of Lexical Taxonomies: The Case of WordNet Top-Level", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[7] Hristea F., "On the Semiautomatic Generation of WordNet Type Synsets and Clusters", Journal of Universal Computer Science, (12/4/02)

[8] Schmid H., "Part-Of-Speech Tagging with Neural Networks.", 15th International Conference on Computational Linguistics, Kyoto, Japan, August 1994.

[9] Turian J.,Wellington B., Melamed I. D., "Scalable Discriminative Learningfor Natural Language Parsing and Translation".

[10] Azzurri Ltd. Japan, http://www.azzurri.jp/en/software/clay/index.jsp, Down loaded on 20th May 2007

[11] http://www.surfpack.com/downloads/DBVA-for-JBuilder-for-Windows/23404.html, Downloaded on 20th May 2007

[12]http://pcwin.com/Software_Development/DBVA_for_IntelliJ_IDEA_for_Windows/index.htm, Down loaded on 20th May 2007

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Computational Modeling in Conceptual Models: Widening Scope of Artificial Life

D. S. Kalana. Mendis[1], Asoka. S. Karunananda[2] , U. Samaratunga[3]
[1]Department of Information Technology, Advanced Technological Institute, Labuduwa, Sri Lanka
[2] Faculty of Information Technology, University of Moratuwa, Sri Lanka
[3] Gampaha Wickramarchi Ayurvedic Institute, University of Kelaniya, Sri Lanka

kalanaatil@mail.com[1] ,asoka@itfac.mrt.ac.lk[2]

Abstract

Artificial Life or Alife is concerned with computational models invented by considering explicit knowledge in biological systems. However conceptual models have not been addressed for Alife due to involvement of informal practising methodologies. This leads to concern with tacit knowledge in conceptual models. This paper presents a research, which is incorporated with a computational modelling in diagnosis of human constitutions in Ayurvedic medicine considered as conceptual model. Further Tacit knowledge is the key issue of knowledge modelling aspect because all knowledge is rooted in tacit knowledge. An Intelligent Hybrid system involved with artificial intelligent techniques, namely fuzzy logic and expert system technology has been used to implement the computational model. The result of the modelling of Ayurvedic domain using fuzzy logic has been compatible with the experiences of the Ayurvedic experts. It has shown 77% accuracy in using the tacit knowledge for reasoning in the relevant domain.

1. Introduction

Artificial Life or Alife is a new, multi-focused field of research. Like theoretical biology, it is concerned with formal, computational models of life, but it has expanded its scope by claiming the possibility of *synthesizing* lifelike behaviors within computers and other artificial media [34]. Traditional biology has been primarily *analytic* and reductionist in outlook, taking an organic whole apart and analyzing its components. Chris Langton introduced the field, Artificial Life complements based on explicit knowledge in traditional biology [36]. This has been classified into *Computational Alife, Robotic Alife* and *Chemical Alife based computational models using explicit knowledge of biological systems. But conceptual models for biological systems have not been addressed due to* involvement of informal practicing methodologies [39,40, 41]. This leads to research tacit knowledge modeling in conceptual model. Here we used diagnosis of human constitutions in Ayurvedic medicine as conceptual model.

Ayurvedic medicine has a very strong bearing on the concept of *Prakruti,* which means nature (natural form) of the build and constitution of the human body. According to Ayurveda the path to optimal health is different for people depending on their *Prakruti.* For individuals the Prakurthi is defined as a combination of *(Vatha, Pittha* and *Kapha*) [6]. A balanced state of the *Tridoshas* makes a healthy and balanced person (Physically and Mentally).

Since we all have different combinations of the *Tridoshas,* The diagnosis of p*akruti* offers unique insights into understanding and assessing one's health. It is not merely a diagnostic device but also a guide to action for good health.. It assesses the, dominance of *Tridoshas* and give advice for preventive and primitive health care. The ancient science of Ayurveda is the oldest known form of health care in the world. According to Ayurvedic classification individuals can be grouped into 7 types of their dominancy of components such as

Vata, *Pita*, *Kapha*, *Vata Pita*, *Vata Kapha*, *Pita Kapha*, or *Vata*

Pita Kapha. One of the main principles in Ayurvedic medicine is based on the importance of individual differences with regard to treatments. Ayurveda has gone beyond mere classification and identified possible diseases for each category of people. In general population, human constitution is combination of *Vata*, *Pita*, and *Kapha*. Recognition of human constituent in Ayurveda, is currently based on a standard questionnaire on subjective criteria based on ancient theories of Ayurvedic scholar *Charaka,* 1000 BC and *Susruta,* 600 BC. Questions in concerned are very much user-friendly and based on conceptual model of Ayurveda [6], which is used for finding constituent type, has probes such as repeating questions and classification of constituent type. This has been used for classification of individuals for many centuries. There has been no research into improve the questionnaire although people have realised that the classification is not acceptable sometimes.

Knowledge modeling is concerned with languages, tools, techniques and methodologies for developing abstract models of some target domain or problem solving behavior. Knowledge modeling technologies - in particular problem solving methods and ontologies - are relevant to many disciplines, including knowledge engineering, knowledge acquisition and knowledge management [11]. All knowledge can be considered as tacit or rooted in tacit knowledge[27]. It is important to investigate the methods available for tacit knowledge acquisition. Since tacit knowledge is embedded in implicit nature, fuzzy logic gives great interest of handling such kind of nature. Further more, this can be considered as a method for tacit knowledge acquisition using fuzzy logic. Despite fuzzy logic has been used for knowledge acquisition in such domains, a large portion of the process is manually operated. XpertRule Knowledge Builder extends [31] the graphical knowledge representation paradigm, established since 1988, by its predecessor XpertRule KBS, to new levels of scalability and flexibility. Although the knowledge acquisition accompanied with the methods based on fuzzy logic, but it is exploited the level of transparency and accuracy due to handling manually constructed membership functions. There is a great issue of dealing with constructing membership functions, especially on determinations about intervals of membership functions. Most of the time the knowledge engineer is expected to do this task which leads to arise questions about system validation. Another knowledge acquisition tool for computer assisted diagnosis of postmenopausal osteoporosis using a fuzzy expert system shell [2] is also seen in a position of a manually operated tool for knowledge acquisition. Although WinProlog LPA [5] gives a toolkit (FLINT) based on fuzzy logic for constructing membership functions effectively, but it appears a

manual method for determining the intervals of membership functions. However, in these approaches fuzzy membership functions are defined in an ad-hoc manner entirely based on expert's knowledge. When domains consist of tacit knowledge, expert's decisions are also not consistent. Therefore, we argue that the use of fuzzy logic for modeling tacit knowledge should be formally supported at least to some extend.

In modeling tacit knowledge of diagnosis of human constitutions in to Hybrid Intelligent systems, we invented a novel approach for computational model of diagnosis of human constitution []. An Intelligent Hybrid system involved with artificial intelligent techniques, namely fuzzy logic and expert system technology has been used to implement the computational model. We primarily used fuzzy logic together with statistical technique of principle component analysis for modelling tacit domains. Existing practising methodology of Ayurvedic sub-domain of individual classification has been considered as tacit knowledge. This has been acquired through a questionnaire and analyzed to identify the dependencies, which lead to make tacit knowledge in the particular domain.

2. Classification of Alife

Alife has been classified into Computational Alife, Robotic Alife and Chemical Alife based computational models based on biological systems [35].

2.1 Computational Alife

The purpose of Alife is to make computationally based models of natural biological systems. The ultimate goal is the realization of life in another medium -- i.e., the computational medium of a computer (`computer life'). This can be achieved, because life is a medium-independent phenomenon,

a question of form or processes, not a specific material that constitutes `aliveness'.

2.2 Robotic Alife

Robots and animats built for technical purposes may behave in a `lifelike' manner. This, however, is a by-product of our interpretation, the behaviour of these systems represents a category quite distinct from the behaviour of the carbon-based biological cells and organisms. The ultimate goal is the creation of autonomous, self-reproducing animats, capable of living a life of their own, adapting to a changing environment and eventually evolving into new species if located in an appropriate environment. A more proximate goal is the creation of robots or animats with the full behavioral capacities of living organisms. If any differences are at all recognized between systems based on biochemical mechanisms and `animat' devices based on human design, eventually in very small scale (`nano-technology'), these differences are not judged to effect the principal possibility of achieving the goal.

2.3 Chemical Alife

This view comprises attempts to make real material systems with lifelike characteristics, eventually as *in vitro* models of prebiotic processes; primitive metabolic systems; the so-called Eigen hypercycle systems, etc.

3. Proposed Approach

We postulate a novel approach enhancing the ability of modeling tacit knowledge in conceptual model by using an Intelligent Hybrid system []. It has been exploited the process of the new approach in following steps.

3.1 Acquiring Knowledge

The approach begins with by acquiring tacit knowledge. This can be done as an interview between domain experts and the knowledge engineer. Using the interviewing process between expert and knowledge engineer, tacit knowledge has been acquired and mapped in to a questionnaire based on Likert scale technology [27], We have chosen to acquire tacit knowledge into a questionnaire since it is more convenient for further analysis. Once tacit knowledge has been acquired then we should analyses the knowledge for finding dependencies. The questionnaire has been analyzed using principle component analysis (PC) to find dependencies [3].

3.2 Principle Components for Tacit Knowledge Modeling (Model Refinement)

Let *S* be the set of all questions in the questionnaire and *P* be the set of all extracted principle components.

$$\text{Further, } P = \{PC_1, PC_2, ..PC_{n-1}, PC_n\}$$
$$S = \{S_1, S_2, ..S_{m-1}, S_m\}$$
$$\Rightarrow$$
$$PC_i = a_{1i}S_1 + a_{2i}S_2 + .. + a_{m-1,i}S_{m-1} + a_{m,i}S_m$$

Let *M* be the principle components Matrix for filtered tacit knowledge.

$$M = \begin{bmatrix} a_{11}a_{12}..a_{1n} \\ a_{21}a_{22}..a_{2n} \\ . \\ . \\ a_{m1}a_{m2}..a_{mn} \end{bmatrix} \qquad (2)$$

$$\therefore PC_1 = a_{11}S_1 + a_{21}S_2 + .. + a_{m1}S_m \qquad (3)$$

$$PC_2 = a_{12}S_1 + a_{22}S_2 + .. + a_{m2}S_m \qquad (4)$$

$$PC_{n-1} = a_{1n-1}S_1 + a_{2n-1}S_2 + .. + a_{mn-1}S_m \qquad (5)$$

$$PC_n = a_{1n}S_1 + a_{2n}S_2 + .. + a_{mn}S_m \qquad (6)$$

For n number of extracted principal components, following computation is concluded.

$$X = \sum_{j=1}^{n} PC_j \quad (7)$$

$$\therefore X = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} S_i \quad (8)$$

Generating Membership Function (Fine–Tuning Analysis)

Let LS be the Likert scale, then

$$LS = [L,..,U] \quad (9)$$

X_L and X_U values are derived from results of the filtered tacit knowledge. It is computed as given below.

$$\therefore X_L = L \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} \quad (10)$$

$$\therefore X_U = \mathrm{U} \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} \quad (11)$$

Let A be fuzzy set defined on a fuzzy concept using the interval of $[X_L,..,X_U]$. Then membership function is as follows.

$$\text{A (X)} = \begin{cases} 0 & X <= X_L \\ (X-X_L)/(X_U-X_L) & X_L < X < X_U \\ 1 & X >= X_U \end{cases} \quad (12)$$

3.4 Fuzzy Rule Base (Reasoning)

Fuzzy rules can be constructed as follows,

Rule 1: If $X <= X_L$ then A (X)=0 %

Rule 2: If $X_L < X < X_U$ then $(X-X_L)/(X_U-X_L)$ %

Rule 3: If $X >= X_U$ then A (X)=100 %

Further, fuzzy rule base can be extended by adding dynamically, in order to function the reasoning process for answers given by the fuzzy rules.

4. Towards modeling of tacit knowledge

The approach has been converted for an implementation using the architecture given below (Figure 1). It is consisted of with modules such as principle component analyzer, database, knowledge base, and fuzzy logic module and inference engine.

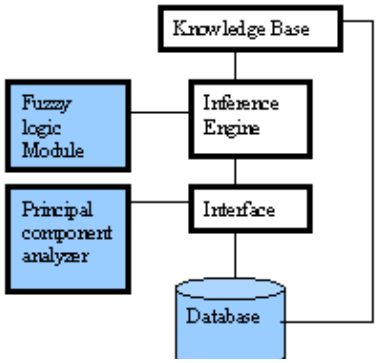


Figure 1: Top-level architecture

Tacit knowledge has been extracted from the expert and formulated in a questionnaire. It is evaluated using Likert scale technology. In the first instance of knowledge acquisition, a pilot survey has been done for the purpose of extracting principal components. The SPSS is used for conducting the functions of principle components extracting.

4.1 Fuzzy Logic Module

The output results of the principle component analyzer would be the input for the fuzzy logic module. In the case of generating membership function, finding the interval is considered as an automated process in this module due to instead of using runtime inputs. This module has been implemented using Visual Basic for widening scope of generating membership function. Further, fuzzy rules have been constructed in the fuzzy logic module.

4.2 Database

Extracted principle components have been stored in Ms Access database, which integrated with the principle component analyzer through the developer interface that is considered as a sub interface of the user interface. The questionnaire consisted of tacit knowledge also been stored in the database that integrated with the user interface.

4.3 Knowledge Base

Explanations for output generated by the fuzzy logic module have been processed using fuzzy rules in the knowledge base. Further, knowledge engineer is given a facility to add new rules in the runtime. The knowledge base has been implemented using FLEX expert system shell, which embedded in WinProlog.

4.4 User Interface

The user interface facilitates for both developer and general user. Once knowledge engineer develops a particular framework for required tacit domain with interaction of the expert, and then general user will be given a facility of using the framework for decision-making purposes. So, it has been divided the user interface in developer interface and general user interface. General user will be able to use a developed framework using a questionnaire, which has been implemented as a web page linked to the database.

4.5 Inference Engine

The inference engine carries out the reasoning whereby the expert system reaches a solution. This is the inference engine of the FLEX expert system shell. Since this is built in to the system there are no development activities with regard to this component in the system. Note that inference engine has nothing to do with the modeling of tacit knowledge but it runs the expert system.

5. Computational model of diagnosis of human constitutions

We have evaluated our approach using Ayurvedic medicine as a domain with tacit knowledge []. In doing so, classification of individuals through clinical examination in Ayurveda has been considered [6]. The clinical examination of Ayurveda is divided into 2 paths, namely: examination through patient and examination through disease.

Prescribing drugs for a disease is depended on both 2 examinations. Classification of individual (human constituents) is included in examination through patient, which defined as a concept called *´prakurti pariksha.´*. Individual can be categorized into *vata* or *pita* or kapha based on the *´prakurti pariksha.´*. It was defined that one type can be dominated but in combination of all 3 types. In the exciting system, the method of analysing constituents is not consistent. Although Ayurvedic practitioners use a questionnaire but leads several problems like dependencies among the questions in the questionnaire and analysis of the constituent type. We addressed these problems to solve using following stages by a computational model.

5.1 Extracting Tacit Knowledge in Ayurveda

In the first instance we mapped tacit knowledge regarding to analysis of constituents to a questionnaire with interaction of an Ayurvedic expert. It is consisted of 72 questions to analyse *vata*, *pita* and *kapha.* It is shown as Figure 2.

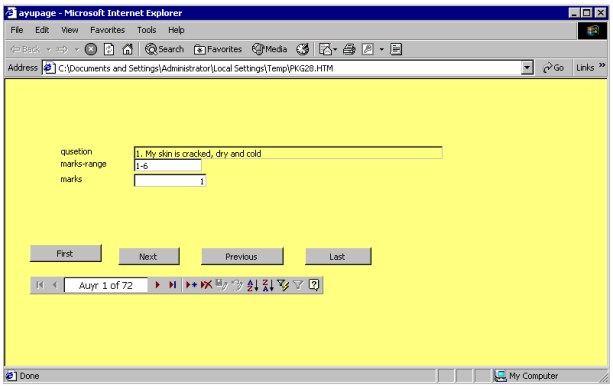


Figure 2: Questionnaire window

no.of students for statistical modeling. Principal component analyzer has been used to remove dependencies. It has been identified 25 principal components using SPSS [17] as shown below in a matrix form. Here V1, V2..V24, K1, k2..K24, P1, P2..P24 denotes question-numbering system in the questionnaire.

$$M = \begin{array}{c} \\ V1 \\ V2 \\ . \\ . \\ V23 \\ V24 \\ K1 \\ K2 \\ . \\ . \\ K23 \\ K24 \\ P1 \\ P2 \\ . \\ . \\ P23 \\ P24 \end{array} \begin{array}{c} \\ V= \\ \\ \\ \\ \\ K= \\ \\ \\ \\ \\ P= \\ \\ \\ \\ \end{array} \begin{pmatrix} 1 & 2 .. & 24 \\ -0.228622 & 0.249362 . & -0.073945 \\ 0.08431 & 0.20654 . & -0.097192 \\ & & \\ & & \\ -0.645803 & 0.232312 . & 0.0067 \\ -0.222147 & -0.06453 . & -0.073514 \\ 0.012511 & -0.096332 . & 0.141314 \\ -0.005642 & 0.268145 . & -0.179992 \\ & & \\ & & \\ 0.409442 & 0.073812 . & -0.115118 \\ 0.696973 & 0.126679 . & 0.098213 \\ 0.430044 & 0.14608 . & 0.023669 \\ 0.243781 & 0.373485 . & -0.040468 \\ & & \\ & & \\ 0.009727 & 0.012529 . & -0.072224 \\ -0.378091 & 0.096985 . & 0.158006 \end{pmatrix}$$

5.3 Analysis Of Human Constituents

Human constituents can be computed in to *vata, pita* and *kapha* in percentages as shown in Figure 3. Membership functions for *vata, pita* and *kapha* have been constructed using the out puts of principle component analyzer.

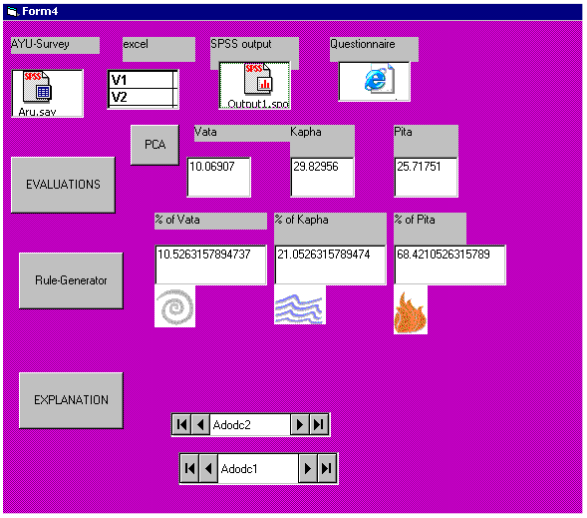


Figure 3: Analysis of human constituents

5.4 *Explanations for Derived Human Constituents*

Possible diseases can be occurred due to dominated constituent type. It is illustrated as shown in Figure 4, which has been implemented through FLEX expert system shell.

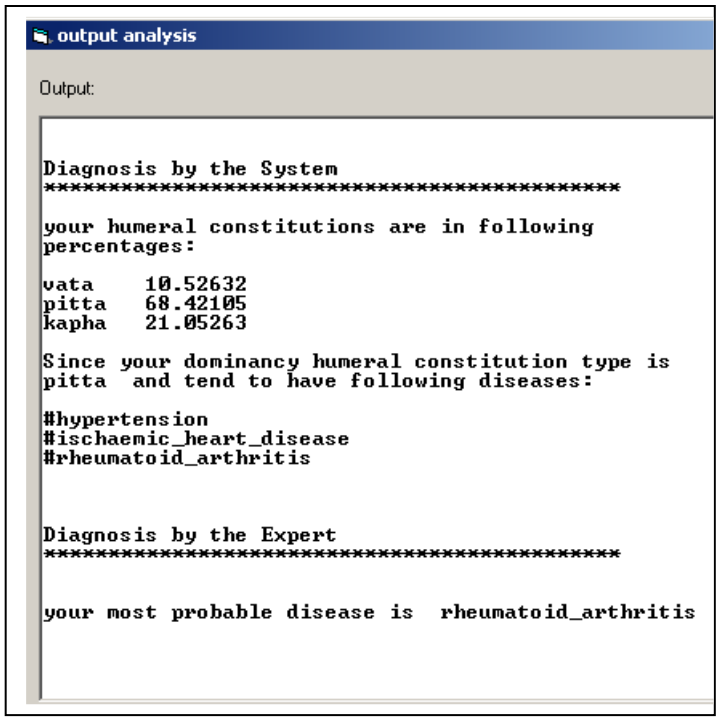


Figure 4: Explanation Window

6. Discussion and conclusion

The expert system developed using this approach was tested with a group of 35 persons of Ayurvedic experts and Ayurvedic medical students. The evaluation was conducted to see far the answers generated by the system matches with the identification by Ayurvedic experts and the students. Further, the system's ability to fine-tune the answers was also tested. It is investigated that 77% of conclusions matches with the system and expert using descriptive statistics.

The system facilitated to derive constituents types in percentages while Ayurvedic experts obtain only the constituent type. As recommendation given by the Ayurvedic experts, determining constituent's types in percentages is an important criterion for prescribing drugs for a disease. Further, our system provide as an option to find out possible diseases. In generally, the system can be used as a self-assessment for finding constituents. According to Ayurvedic medicine, regiments can be done easily by knowing the constituent type. The human constituents can be computed as a combination. So it would help to find the effectiveness of minimum type in a diagnosis.

In doing so, a novel approach was formulated namely, tacit knowledge of experts can be acquired via questionnaire and informal interviews, statistical techniques of principle component analysis (PCA) can be used to identify preliminary dependencies in acquired tacit knowledge and fuzzy logic with PCA as the input can be used for developing a model for modeling tacit knowledge. This leads to present a formal practicing methodology for tacit knowledge modeling in conceptual models. It has been achieved a computational model for diagnosis of human constitution. This leads to convince of computational modeling using conceptual models in Alife.

6. References

[1] Bellman and Zadeh, (1970). R.E. Bellman and L.A. Zadeh , Decision making in a fuzzy environment. Management Science 17, pp. 141–164.

[2] Binaghi E., De Giorgi O., Maggi G., Motta T., Rampini A (1993), *A knowledge acquisition tool for computer assisted diagnosis of postmenopausal osteoporosis using a fuzzy expert system shell*, Computers and Biomedical Research, 26,pp.498-516.

[3] Chatfied C(1996)," Introduction to Multivariate Analysis", Chapman and Hall.

[4] Coppin G, Skrzyniarz A (2003), "*Human–centered processes : Individual and distributed decision support*", IEEE Intelligent systems, pages 27–33.

[5] Dave Westwood, Flex reference guide, LPA, U.K

[5] Dubey G.P , (1978)The Physiological concepts in Indian medicine, Science and Philosophy of Indian medicine, Shree Beldyanath Ayurved Bhawan Ltd.

[6] Evangelos Petroutsos,(1998), Mastering Visual Basic 6, BPP Publications, New Delhi

[7] George J (1997), "Fuzzy sets and Fuzzy Logic", Prentice Hall of India.

[8] George J.K, Yuan B.(1995), Fuzzy sets and Fuzzy logic, prentice hall of India, pp. 280–300.

[9] Hayashi Y, Tazaki E, Yoshida K, Dey P (1988), "Medical diagnosis using simplified multi-dimensional fuzzy reasoning", In: Proceedings of

IEEE 1988 International Conference on Systems, Man and Cybernetics, Beijing, China, 1988. p. 58–62.

[10] http://kmi.open.ac.uk/knowledge-modeling.html

[11] Jonson L (1988), Expert system Architectures, Kopan Page Limited.

[12] Karl B, Klaus P, A,Yoichi H, Thomas E. Rothenfluh and Harald L (2004), "Knowledge acquisition in the fuzzy knowledge representation framework of a medical consultation system" International Journal of Artificial Intelligence in medicine, pages 1–26.

[13] Karunananda A.S (2000), How To Do Research Author Publication.

[14] Kolousek G (1997), ″The system architecture of an integrated medical consultation system and its implementation based on fuzzy technology", Doctoral thesis, Technical University of Vienna, Austria.

[15] M. Dzbor, 2000. Explication of Design Requirements Through Reflection on Solutions. In Proc. of 4th IEEE Conference on Knowledge-based Intelligent Engineering Systems. Brighton, UK, pages 141-144.

[16] Matei Ciobanu Morogan, (1997), SPSS for windows, release 8.0.0, DSV KTH⁄ SU, Sweden.

[17] Mendis D. S. K., Karunananda A. S. and Samarathunga U. (2004), *An Expert system for analysing Aurvedic human constituents,* Shamisha – Journal of Aurveda

[18] Mendis D. S. K., Karunananda A. S. and Samarathunga U. (2004), *Multi-Techniques Integrated tacit knowledge modelling system*, International Journal of Information Technology, Vol 9, pp 265-271

[19] Mendis D. S. K., Karunananda, A.S. and Samarathunga U. (2003), *Framework for modelling of tacit knowledge – Case study using Ayurvedic domain,* In proceeding of the International Information Technology Conference (IITC'2003), pp 76-82, Colombo, Sri Lanka

[20] Mendis D.S.K, Karunananda A.S, Samaratunga U. (2003), "*Reasoning with Uncertainty*", Health Informatics Conference, Colombo.

[21] Mendis D.S.K, Karunananda A.S, Samaratunga U. (2003), "*Tacit knowledge modeling*", (2003), Sri Lanka Association for Artificial Intelligence Annual Session and Annual General Meeting, Colombo.

[22] Morrison D.F(1990), "Multivariate Statistical Methods", Mc Graw-Hill, Inc. PP. 312 - 357.

[23] Noak V(2000), "Discovering the world with Fuzzy logic", A Springer – Verlag Company, PP. 3 – 50.

[24] Peter J. (1995), Introduction to expert systems, Pearson education Asia, pp. 182–199.

[25] Rajeev K (1995), "Integrating knowledge bases in expert system shells: an open system approach", International Journal of computer application in technology, pages 78–89.

[26] Richards D. and Bush P., "Measuring, Formalizing and Modeling Tacit Knowledge" IEEE/Web Intelligence Conference (WI-2003) Bejing.

[27] Ross Dawson, (2001), "Developing Knowledge-Based Client Relationship ", Butterworth Heinemann.

[28] Senkar J (1995), ″Network and tacit knowledge in innovation″, Economics et societies, pp. 99-118.

[29] Tripathi S.N(1978), "Clinical Diagnosis", Science and Philosophy of Indian medicine.

[30] XpertRule Knowledge Builder, www.attar.com.

[31] Zadeh L. A (1988),Fuzzy Logic, Computer IEEE, pp. 83-93.

[32] Zimmerman H.J (1985), Zimmerman Fuzzy set theory and its applications, Kluwer–Nijhoff, Boston

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Ontology Driven Approach to Disaster Management

A. Thushari Silva & Ajith P. Madurapperuma

Faculty of Information Technology,
University of Moratuwa, Sri Lanka.

E-mail: {thushari,ajith}@itfac.mrt.ac.lk

Abstract

Disasters have become a common threat to humans. Although disasters cannot be evaded, human can manage the disastrous situations to minimize the damage. A successful disaster management system requires information from heterogeneous data sources that belongs to various services. As such, data integration is a crucial issue for disaster management systems. Recent research has recognized ontology as a potential approach to develop solutions for systems involving diversified sources of data. This paper presents an ontology driven approach to data integration in disaster management systems. The proposed ontology enables editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. Protégé ontology development environment has been used to implement the proposed disaster management ontology.

Keywords: Disaster Management System Ontology, Data Integration

1. Introduction

Environmental conditions in the modern world are changing in an irregular manner world wide. This has resulted in the occurrence of natural disasters as well as for the disasters for which humans are responsible. For example, the Sumatran-Andaman earthquake that triggered a Tsunami in 2004, the earthquakes which occurred in Kashmir/Pakistan in 2005, the hurricanes Katrina and Rita in 2005, and the outbreak of SARS – a condition that emerged in southern China in 2003 that spread rapidly throughout the world, are a few that shocked the world unexpectedly.

As such, prevention of disasters and management of disastrous situations have become a theme of common interest. One of the biggest issues in disaster management can be identified as the need for integrating and coordinating many services to handle disastrous situations. For example, disaster management requires considering various services including hospitals, community centres, police, ambulance and funding agencies. Obviously, modern information and commutation technology can provide potential solutions for integrating these services. However, effective integration of services scattered over heterogeneous hardware and software environments require technologies that can bring all services together. According to modern researchers [3, 4, 11, 9, 20, 10, 18] ontology has been identified as a technology which forms categories of objects (services) which are related to each other. The applications of ontologies have become so challenging, as data related with different services are presented in different formats. As such, ontologies become an artefact which facilitates the communication among heterogeneous sources of information. Therefore, it is appropriate to devise an ontology driven approach to design and implement disaster management systems that can link up associated services together.

This paper proposes an ontology, for the design of disaster management systems that has access to data sources available in different formats including XML, Text, graphics, and databases entries. The disaster management ontology enables editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. Open source software, Protégée has been used to develop the disaster management ontology.

The rest of the paper is organized as follows. Section 2 is a review of the current state of disaster management systems and identifies the need for service integration as a major issue of such systems. Section 3 points out the relevance and the appropriateness of the use of the concept of ontology to integrate services in disaster management systems. Section 4 describes the architecture and functionality of disaster management ontological systems. In the fifth section we discuss disaster management ontological systems can be further improved in order to provide better access and functionality.

2. Overview: Disaster management system

Disasters can occur in forms such as Tsunamis, Hurricanes earth quakes and terrorist's attacks. Unexpected and unavoidable disasters are generally and may create scenarios that are novel. This leads to generate, or gather unstructured or semi structured data at different locations in an unplanned manner resulting in heterogeneous and autonomous data sources. Furthermore, disaster management cannot be an isolated application. It must quickly be aware of information from many sources which are autonomous and heterogeneous. The data and information that we keep in a disaster information management system will have to be timely, reliable, consistent and secure. One of the main observations during the post tsunami relief and rehabilitation stage was that different organizations collect and read data in using a variety of structures, formats and management tools. Thus, integrating such data, which allows isolated data sharing by giving meaningful access to such data, has become crucial due to the heterogeneity of those data sources.

The world looks at a disaster, often willing to help, but needing an accurate picture about the current situation. Possible funders, including the general public, need to see what is necessary. Relief organizations need an accurate on the ground picture so that resources are sent to the right place and are not wasted. Families of those involved need survivor information. For chemical or biological events, scientists across the globe with expertise in a particular field may need to apply themselves to the problem at short notice.

It is evident that, once a disaster occurs, the people who are dealing with disaster management seek accurate real time data. In order to provide such information there should be unique access to those autonomous and heterogeneous data sources. This exploits the concept called data Integration. Semantec heterogeneity makes integration a challenging task and dynamic changing of such sources makes the process further complicated. Once we resolve the semantic heterogeneity, multiple, autonomous data sources will be able to be amalgamated and to provide a unique interface to users of disaster data. As described in the next section, the ontology by its definition has the power of resolving semantic issues.

A few Disaster Management Systems have been developed recently. 'Sahana' [17] is one such disaster management system which is already developed and deployed in different countries as a stand alone application. Sahana Disaster Management System is free and open source software developed using LAMP (Linux-Apache-Mysql-PHP/Postgres) environment. Providing a wide-ranging solution for recovery and rehabilitation is one of the main objectives of the Sahana. This has several modules including missing persons registry, inventory control system, camp management. Idea of integration of data with other sources has not well developed Sahana as well as other tools.

The current disaster management systems provide users with the access to pool of data/information without any structural guidance for navigation through a huge collection of resources. In other words, such systems cannot be used by ordinary persons who are not educated about related services. More importantly, since people are excited and strenuous in disastrous situations, it is rather impractical for them to explore various services consciously. Therefore, a technology that can integrate various services and guide the users to explore services in a distributed environment is a crucial for implementing effective solutions for disaster management. The recent research shows that the concept of Ontology is the best technology for implementing integrated solutions for problem solving in heterogeneous environments [10]. Section 3 gives an overview of ontology.

3. What is Ontology?

People have evolved from procedural thinking to object oriented concepts as a means of modelling the real world. However, object-oriented thinking encourages looking at the world through objects, which are specifically defined and separated from each other. In the recent past, ontology has emerged as an approach to look at the world though categories of objects which are related to each other in some manner. However, ontology is still a relatively new area in computing. In fact, there is no exact definition for ontology, and many views are expressed to define ontologies. There are several working definitions for this valuable term ontology starting from "an explicit specification of a conceptualization" [3, 4]. In the time of writing this paper there are more than seven definitions some time contradictory [15] for ontology [2].

The spirit of ontology is portraying of concept in the world which we live. Most of the scientists in the computer world use the term *class* instead of *concept* when defining ontology. In computing terminology ontology behaves as a controlled vocabulary which consists of classes of objects, classes of relationships

among these objects and consists of grammar which has enough capability to specify rules for using vocabulary terms in meaningful manner in the domain of interest.

In spite of some puzzlement about the definition, numerous practical developments in different areas have taken place during the past few years. For example, a system for specifying ontology in the form which is compatible with multiple representation of languages called Ontolingua [4,5] and a framework for finding data in distributed geographic information services [10]. OIL [8] has been developed as the first Ontology representation language to meet W3C standard for Semantic Web and, very recently, and Web Ontology Language (OWL) emerged as the key ontological language for Semantic Web applications [13].

From the database system point of view, ontologies also behave like conceptual database schema. The main functionality of a conceptual database scheme is to provide a logical description of shared data at the logical level, allowing application programs and databases to interoperate without having common sharing structure. While a conceptual schema defines a relation on *data*, ontology defines terms which with to represent *knowledge*. It is evident that there is an astonishing relationship among relational data bases and ontologies. In addition, as highlighted in the previous section disaster data which are in different formats such as relational Databases, XML, and text can be integrated to provide a unique interface by solving problematic level heterogeneity by using ontology. Such ontology is capable to extract the data from those autonomous & heterogeneous data sources and provide intelligent answers based on the requested query.

There are many tools for construction of ontolgies. Some tools are specific to development of ontologies, while others are used for editing, merging, annotating of ontologies. OntoWeb has identified 11 environments, namely, Apollo, LinkFactory, OILEd, OntoEdit, Ontolingua, OntoSaurus, OpenKnoME, Protégé 2000, SymOntoX, WebODE and WebOnto for building or development of ontologies. All these are graphical tools, based on object- oriented concepts and various extensions to XML including RDF, RDFS, OIL and DAML+OIL. Most tools are developed using Java, while some have chosen C++. Further almost all tools provide a library of ontologies. Most tools have some means of working with other Ontology environments, which support interoperability among tools.

Protégé is the latest tool in the ontology development environment [16]. Since Protégé-2000 provides interactive knowledge-base-development environment and ontology-design frame work, the experts can perform knowledge management-task efficiently. It has the capability to handle knowledge bases in multiple languages. Due to this feature, ontology developers can access relevant information quickly whenever they need it, and can use direct manipulation to pilot and manage ontology. Protégé has three types of plug-in libraries; (1) **backends –** provide user with store and import knowledge bases in various formats. Ontlogies in various formats including RDF Schema, XML files with a DTD, and XML Schema files can be imported ; (2) **slot widgets-** which are used to display and edit slot values or their combination in a domain-specific and task-specific ways, and (3) **tab plugins**- which are knowledge-based applications usually tightly linked with Protégé knowledge bases.

Some other systems translate a common ontology into several systems. One such tool, Ontolingua [4,5] was developed with a set of portable ontologies as a means for share and reuse formally represented knowledge. It deals with ontologies represented in predicate logic, frames and relational languages. The ontology of Ontolingua defines forms with classes, relations, functions, objects and theories. Ontolingua ontologies can be translated into different languages including KIF [6], LOOM [14], Epikit [7] and EXPRESS [19]. Using Ontolingua the same ontology can be used for different purposes with the help of KIF, LOOM, Epikit and EXPRESS. For example, LOOM can be used for conceptual design of the ontology and managing the knowledge base of facts and objects. Epikit provides explanatory reasoning. EXPRESS is the standard language for logical database design for sharable data. KIF provides ontologies for sharing data.

Although Ontolingua works with four other systems, it is a relatively bulky product [8] with an essentially narrow range of applications. Until a framework for the development of a generic top-level ontology is created and translation of Ontolingua ontologies into a wide range of ontological systems is enabled, its usage will be very limited. In its present state Ontolingua cannot be used as a general system to support ontology translation and sharing for the Semantic Web application. Nevertheless, the

Ontolingua approach can be considered as a good initiative and the same philosophy can be extended towards the development of a more general approach to translating ontologies across various applications.

4. Ontology for disaster Management

The design of a disaster management ontology is based on the identification of data sources that should be considered and the ontology development environment, which will be used to construct the ontology. Our research has shown that a disaster management system concerns with services such as security service, health service, emergency service and charity service. The study also revealed XML, databases or text files as the main formats of data sources. Therefore, we have decided to develop the disaster management ontology that can work with XML and Databases technologies.

There are number of design goals that should be fulfilled by the proposed ontology, in order to perform operations including editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files, and guide users in disaster environment while linking available services in the disaster management. Most important goal is, integrating textual data and multimedia data. It is obvious that, unexpected nature of the disaster creates a lot of data which are unstructured and stored in text format. Integrating textual data is difficult, as objects and relationship among the objects is hard to extract from natural language. To circumvent such difficulties, a component called 'intelligent information extractor' which resolves textual heterogeneity while dealing with reference ontology, is to be developed. The next goal is to sustain flexibility of the disaster management ontology. In other words, once the system was developed it should be extended by integrating other newer data sources in the disaster domain. Then only data integration can be achieved successfully. This challenge has been addressed by providing separate wrappers for each data source. Thus only a new wrapper is needs to be developed in case of integrating with new source. Third goal is providing automatic integration. Since extracted data from the underlying data sources should be available as fast as possible, ontology is developed to support automatic integration.

A survey on tools and environments for ontology development has shown that Protégé is the most widely used open source software for development of ontolgies. Therefore we have decided to use protégé for the design of the disaster management of ontology. As required by the Protégé system, different data formats will be converted to what is called Resource Description Framework (RDF). RDF [1, 12] provides a means for expressing semantics of documents without making any assumptions on the structure of the document.

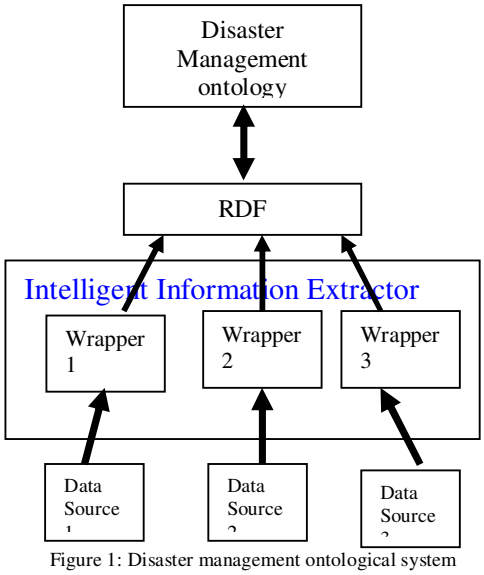


Figure 1: Disaster management ontological system

Figure 1 shows the top level design of the disaster management ontological system. The overall approach can be seen as four-layer architecture with ontology layer, RDF layer, wrapper layer and the data source layer.

A. Ontology layer

Ontology Layer which contains disaster management ontology is the nucleus constituent of the system. Its functionality includes editing, sharing, converting, searching and querying on data available in various formats in databases, web servers, text files and image files. In other words ontology is knowledgeable about the disaster domain including all services and it deals with external data sources and gets real time instances of such services.

The RDF layer construct the RDF data for each and every external source which contains different data formats including XML, relational and text and resolves the semantic heterogeneity among them. Developed ontology has been provided with enough strength to integrate underlying data sources to provide unique access over them and provide an intelligent answer based on such data sources.

Most importantly, Disaster Management ontology has been developed in such a way that a user of this system is guided by the ontology itself for navigating through related services. This, automatic navigation through related services including security service, health service and charity service fulfils the need of assistance for decision making, for any individual in such disastrous situation. For an example consider a child who missed his/her parents wants to find them. It has been noticed that any information about them has not been recorded among the disaster data. In that case child may not know what to do next? In this scenario our ontology can guide him to go for the hospital services before going to the charity service.

Protégé ontology development environment has been used to implement the proposed disaster management ontology. The protégé back end plug-in allows knowledge bases to be in different formats. Using this feature heterogeneous data sources can be plugged with developed ontology after resolving heterogeneity among those sources.

B. RDF Layer

This layer consists of RDF form for each data source which has been generated by the wrappers. The main functionality of this layer is to resolve Semantic heterogeneity in the extracted RDF formats. The translated ontology may differ from each other and may not share the same vocabulary even for referring to the same entities. For example, in case of searching for a person may use the term person in its ontology, while dealing with the donations, it may use the term person to refer to individual who has donated. This simple example shows that there is a need for translation or merging between ontologies, before making any attempt at machine interpretation of meaning. Consequently ontology merging operation is to be performed in this layer.

C. Wrappers

Wrappers are fundamentally parsers which extract the data from different sources and resolvs syntactic differences. In this layer, source-specific wrappers have been developed to transform each and every data source which contain data with different formats, into common RDF. Source-specific wrappers are developed to provide the flexibility for integration of other data sources as well. Integration approach has been developed by establishing a joint data format all over the underline data sources. As for the semi-structured data format, RDF has been chosen. The key point of our approach for integration is we convert all other different data formats into RDF data using reference ontology, and only performs queries locally to the resulting data stored in the central repository without going again to the data source.

D. Data source layer

Data sources are coming from heterogeneous systems of hardware and software. Mainly we are considering data sources available in different formats including XML, Text, graphics, and databases entries. Thus disaster data in such sources may be structured or semi structured. In case of unstructured or semi structured refinement over underlying data sources are to be performed.

5. Discussion

We have been involved in the development of disaster management ontology. Once the wrappers are constructed they will be integrated with disaster management ontology. Then this ontology can be used to query about disaster management information coming from various data sources for which wrappers are developed. It should be noted that this project is more than writing wrappers and converting data sources to RDF. More importantly, we have constructed the ontology in such a way that a user of this system is guided by the ontology itself for navigating through related services. This feature address the issue that disaster management ontology developed for a country cannot be used by another country due the differences in the services available.

This ontology has many advantages it can provide intelligent answers for any query. For instance suppose a user want to find a person who went missing during a disaster and Information shows that there is no particular information about that missed person. Then the developed ontology provides guidance about what he should do next while automatic connection to corresponding services. This ontology can be used by any user who does not have very good knowledge in IT. We deliver an ontology running on a web server and can be accessed by PC.

Further work of this project has many directions. Firstly, this ontology can be extended to enable drawing from new data source just by developing a wrapper for the particular data source. However, the above data formats are very common in use and they are adequate for extracting data from many sources.

Secondly, from maintenance view point, the overall system will be improved to edit, merge, share, and search through expert system. From this end, if we merge with an ontology which describes Geographical Information Systems we will be able provide an answer for queries including geographical information.

Thirdly, we will enhance the power of developed ontology while enable accessing of the system by using mobile devices. This exploits any person from any place at any time can access this developed ontology to get any information.

5. References

[1] Frank, M. and Eric, M. 2004. "RDF Primer", W3C Recommendation, 10th February 2004, http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

[2] Guarino, N, "Understanding, building and using ontologies", International Journal of Human -Computer Studies, vol. 46, pp. 293-310, 1997.

[3] Gruber T.R, "Towards principles for the design of ontologies used for knowledge sharing", International Journal of Human-Computer Studies, vol. 43, pp. 199-220, 1995.

[4] Gruber T. R, "A Translation Approach to portable Ontology Specification", Knowledge Acquisition, pp 9-220, Apr 1993.

[5] Gruber T.R, "Ontolingua: A mechanism to support portable ontologies", Technical Report KSL pp 91-96, 1992

[6] Genesereth, M. R. and Fikes, R. E.. "Knowledge Interchange Format, Version 3.0 Reference Manual", Technical Report, Computer Science Department, Stanford University Logic-92-11, 1992.

[7] Genesereth, M. R. "The Epikit Manual", Epistemic, Inc. Palo Alto, CA,1990.

[8] I, Horrocks, D. Fensel, J. Broekatra, S. Decker, M. Erdmann, C. Goble, F. Van Hermelen, M.Klein, S. Staab, R. Studer, and Motta, "The Ontology Inference layer OIL",Technical Report,2000.

[9] Jin Y., Decker S., Widerhold Gio, "OntoWebber: Model-Driven Ontology-Based Web Site Management", Semantic Web and Database, 529-54, 2000.

[10] Klien E., Mmichael L, Werner K, "Ontology-Based Discovery of Geographic Information Services", unpublished.

[11] Lenat, D., Miller, G. and Yokoi, T.. Cyc, "WordNet, and EDR", Communications of the ACM, Vol. 38, pp 45-49, 1995.

[12] Lassila, O. and Swick, R.. "Resource Description Framework (RDF)", W3C Recommendation, 1999 http://www.w3.org/TR/WD-rdf-syntax/

[13] McGuinness, D. L. ,van Harmelen, F. "*OWL Web Ontology Language Overview"*, W3C Recommendation 10th February 2004.

[14] MacGregor R. "The Evolving Technology of Classification-based Knowledge Representation Systems, in: Principles of Semantic Networks: Explorations in the Representation of Knowledge", edited by Sowa, J.F., San Mateo, CA: Morgan Kaufmann,1991

[15] Noy N. F. and Deborah L. M, "Ontology Development 101: A Guide to Creating Your First Ontology", Technical Report,2001, http://www.bibsonomy.org/bibtex/208759c07a29e371220cddb39a26a6200/fparreiras.htm

[16] Noy, N F, Fergoson, R. W. and Musen, M. A., 'The Knowledge model of Protégé 2000: combining interoperability and flexibility", Stanford medical Informatics Technical Report, Stanford university, 2000.

[17] Rachid L., Careem M., Silva C., Silva D,, Weerawarna S., "Sahana: Overview of a Disaster Management System", International Conference on Information and Automation, pp. 15-17, 2006.

[18] Shamsfard, M. and Barforoush, A. A. "Learning ontologies from natural language texts", International Journal of Human-Computer Studies, Vol 60, pp 17-63, 2004.

[19] Spiby, P., "EXPRESS Language Reference Manual", International Standard ISO 10303-11, Project Three ("Express Language")

Working Group Five ("STEP Development Methods") of Subcommittee Four ("Manufacturing languages and Data") of ISO Technical Committee 184 ("Industrial Automated Systems and Integration"); International Standards Organization. 1991.

[20] Warren, D. H. D. and Pereira, F. C. N., "An efficient easily adaptable system for interpreting natural language queries", Computational Linguistics, Vol. 8, pp 110-122, 1982.

Sri Lanka Association for Artificial Intelligence (SLAAI)
Proceedings of the fourth Annual Sessions
31st October 2007 - Colombo

Intelligent Elevator Group Control System

T. K. W. Senevirathna, H. Ekanayake and T. N. K. De Zoysa

University of Colombo School of Computing
Colombo, Sri Lanka
kanchana.senevirathna@gmail.com, {hbe,tnk}@ucsc.cmb.ac.lk

Abstract

High rising building is a common sight in most of the cities today. Fast and efficient elevator transportation is a key feature when creating these kinds of buildings. Lot of research has been carried out to build an intelligent system that satisfies the need of elevator control as it is a must to have intelligent elevators in the future. This paper proposes a destination control system for elevator group controlling which fully utilizes destination information. Fuzzy logic concepts are used to enable the elevator control system to make decisions. The design criteria include of optimizing movement of elevators with regard to several factors such as waiting time, riding time, energy, load, etc. Software simulation is done in order to capture the performance of the proposed system with compared to conventional approaches.

1. Introduction

Since the start of elevator systems, each and every one carried out the same operations just by using up and down hall call buttons and giving the destination inside a car, not allowing that much of processing of data to be done. However, this type of system provides some disadvantages such that the supervisory system of the elevators does not receive information on the destination of the passengers before they board the car. Consequently, car assignments are based on far less than 50% of the traffic information that passenger could supply earlier to the system and therefore assignments are obviously poor in quality. Such inherent disadvantage present in conventional system can be improved by using a destination oriented system.

The choice of optimization target in the cost function is important when considering the overall elevator performance and the service level. The most general optimization target in group controls has been the minimization of the average and maximum hall call times. Recently the cost functions have become more comprehensive. Instead of one target, multiple targets are optimized. A number of costs, such as call time, passenger waiting and journey times, car load factor, energy consumption, transportation capacity and number of starts, can be considered during the call allocation. When optimizing one target, the other features may suffer. For instance, when optimizing the energy consumption, the passenger waiting times may increase. Several optimization targets can be optimized within one control if the most suitable target is switched or prioritized according to the prevailing traffic pattern or requests.

Fuzzy logic [3][4] is used as an intelligent approach to optimize multiple targets such as waiting time, riding time, load and distance. This system determines the optimum car for a particular request. System requires the passengers to enter their destination before entering to the car, using a numeric keypad which is located in the requesting floor. The system then assigns the passenger to a car and displays the assigned car back to the user. The system is tested in simulating the traffic conditions of 3 cars in a 15 floor building.

2. Conventional Elevator Group Controller Architecture

A typical architecture of an elevator control system [1] is shown in figure 1. The control boards of an elevator group are usually situated in a machine room. The group control is the "brain" of the elevator system, deciding where the cars should go and stop. There can be one or several group controls in an elevator group. One of the group controls is the master that delivers the hall calls to the elevators, and the other group controls are backups. Other functions inside the car, e.g. registering and canceling of car calls, door control, and measurement of the car load, are handled by the elevator control. Using the latest microprocessor technology, part of the elevator control has been distributed among the elevator components. "Intelligent" elevator components communicate with each other through serial

transmission using Control Area Network (CAN)[5]. Modern elevator controls provide built-in elevator monitoring devices or remote building monitoring systems to follow the elevator traffic. Typical control software for an elevator component includes an operating system, task-scheduling programs, input, output and communication programs, and programs for controlling and optimizing the function of the component.

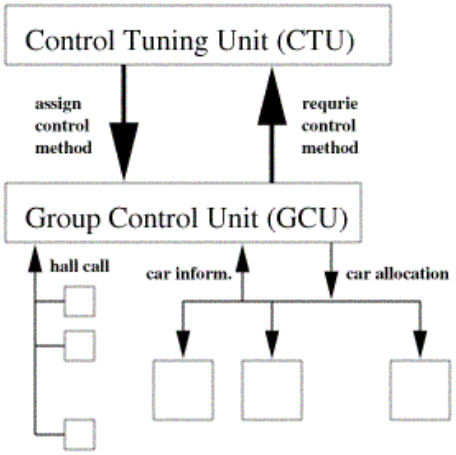


Figure 1: Architecture of conventional Elevator Group Control System

2. Design Criteria

The elevator configuration is chosen so that the generality of the problem is preserved and the simulation can be visualized which means that the building should not have too many or too less, number of floors and the number of elevators should be such that they operate smoothly under varied traffic intensity. Here, the number of elevators was fixed to stress more on the group control of elevators rather than the selection of optimum number of elevators. Thus 15 floor building with 3 cars is selected for testing and simulation. Since the system needed destination information of a passenger before entering the car, that information is fed to the system with a numeric keypad located in every requesting floors.

The objective is to design a dispatching logic that achieves desirable results with respect to

- Minimize the waiting time of a passenger.
- Minimize the riding time of a passenger.
- Minimize the crowding of an elevator.
- Minimize the traveling distance of each elevator.

Also it assumes that a stop of an elevator will take 3 times the time taken to travel one floor. Maximum load for each car is assumed to be 20 persons.

3. Related Work

Although it is hard to implement and compare with most of other new approaches, following points gives a comparison with adaptation of the concepts by the proposed algorithm regarding some specific features.

3.1 Zoning

Zoning [5] has been a planning strategy from the start of high rising building elevator planning. Each car is assigned a zone of the building. It answers hall calls within its zone. The goal of the zoning approach is to keep the cars reasonably well separated and thus keep the interval down.

In fuzzy logic approach calculation of hall call area and destination call area helps to achieve dynamic zoning. If the call area is near to 0, algorithm will give a high priority to the lift when selecting the lift. This concept is further developed with a destination controlling system since it has the destination information before assigning a lift to a request.

3.2 Search based approaches

Another control strategy is to search through the space of possible car assignments, selecting the one that optimizes some criterion such as the average waiting time. Greedy search strategies [5] perform immediate call assignment, that is, they assign hall calls to cars when they are first registered, and never reconsider those assignments. Non-greedy algorithms postpone their assignments or reconsider them in light of updated information they may receive about additional hall calls or passenger destinations. Greedy algorithms give up some measure of performance due to their lack of flexibility, but also require less computation time.

Approach used in this project can be considered as a greedy approach. It assigns a car immediately when a request is made. Although this has the disadvantage of greedy algorithms, i.e. lack of flexibility, this has been overcome by various factors. Some of the factors are getting destination information before getting into a lift contrastingly to conventional systems and reducing psychological waiting time by displaying assigned lift earlier. (This is simulated in the simulation software) Also reassignment of lifts will result badly when it comes to dynamic zoning. Also avoiding non greedy approaches may result in less computation thus will be a less complex algorithm which is easy to understand and further develop.

3.3 Adaptive and learning approaches

Some of the optimization approaches use neural networks for learning traffic pattern to select the best rule set for fuzzy logic, to optimize parking [7], etc. However there are many situations in which training examples are costly or even impossible to obtain. Reinforcement learning [8] is more applicable in these more difficult situations where the only help available is a 'critic' that provide a scalar evaluation of the output that was selected, rather than specifying the best output or a direction of how to change the output. But in the applications of reinforcement learning or any other learning approaches it needs a large set of data to be trained on. This should be concerned regarding the application of real elevator system, since one would want to perform the initial training in simulation in any case, not only because of the large amount of experience needed, but also because performance would be poor during the early stage of training. Even though training will be done in a simulation, traffic generation of a simulator may not be accurate. Also elevator traffic depends so much on the type of the building [2]. Thus learning approaches are too much effort when building an algorithm for elevator group controlling. But this approach can be used for forecasting traffic and parking of elevators which will be a future addition to proposed algorithm in this project, which describes only immediate call allocation mechanism

4. Fuzzy controllers: An overview

Fuzzy controllers, [3] [4] contrary to classical controllers, are capable of utilizing knowledge elicited from human operators. This is crucial in control problems for which it is difficult or even impossible to construct precise mathematical models, or for which the acquired models are difficult or expensive to use. These difficulties may result from inherent nonlinearities, the time varying nature of the processed to be controlled, large unpredictable environmental disturbances, degrading sensors or other difficulties in obtaining precise and reliable measurements and a host of other factors. It has been observed that experience human operators are generally able to perform well under these circumstances.

The knowledge of an experienced human operator may be used as an alternative to a precise model of the controlled process. While this knowledge is also difficult to express in precise terms, an imprecise linguistic description of the manner of the controller can usually be articulated by the operator with relative ease. This linguistic description consists of a set of control rules that makes the use of fuzzy propositions. A typical form of these rules is exemplified by the rule.

If the *temperature* is very high
AND the *pressure* is slightly low
THEN the *heat change*
should be slightly negative,

Where temperature and pressure are the observed state variables of the process, and heat change is the action to be taken by the controller. The vague terms very high, slightly low and slightly negative can be conveniently represented by fuzzy sets defined on the universe of discourse of temperature values, pressure values and heat changes values, respectively. This type of linguistic rule has formed the basis for the design of a great variety of fuzzy controllers described in the literature.

A general fuzzy controller consists of four modules: a fuzzy rule base, a fuzzy inference engine and fuzzification/ defuzzification modules.

5. Applying fuzzy logic to Elevator control system

In this system destination details are obtained before a passenger load into the car through a numeric keypad located in the floor we can use these additional knowledge when applying fuzzy logic. For fuzzification 6 base variables has been identified: 5 input variables and 1 output variable. Following quantities are given as input to the Fuzzy interpreter:

- **Waiting time** - Total time an elevator needs to travel from its current position to the new hall call.
- **Riding time** - Total time a passenger spent in the elevator until he/she reached at his/her destination.
- **Loading** - Number of passengers in an elevator.
- **Hall call area** - The area weight of the elevator which goes to the floor where a new hall call is generated.
- **Destination call area** - The area weight of the elevator which goes to the floor where the destination of the new hall call is generated.

"Priority" is the output of the fuzzy controller and the elevator with highest priority value will be assigned for the given request.

5.1 Fuzzification and Fuzzy Reasoning

Each fuzzy system [6] is realized in the form of fuzzy rules as in the following example**:**

Rule 1: If X is A_1 and Y is B_1 then Z is C_1
Rule 2: If X is A_2 and Y is B_2 then Z is C_2

Where X and Y are variables of the condition part, and Z is the variable of the action part. A_i, B_i and C_i are fuzzy parameters characterized by membership functions.

The condition parts of control rules make use of measurements which are usually real numbers. e.g. x^0 and y^0 are matched to their corresponding fuzzy variables by determining their membership values defined as figure 2.

Suppose that X = x^0 and Y = y^0, the reasoning is derived as follows:

- Define the linguistic variables as described in fuzzification section.
- Compute firing levels by mathematical interpretation of these rules, as follows:

  $\alpha_1 = A_1(x^0) \cap B_1(y^0)$
  $\alpha_2 = A_2(x^0) \cap B_2(y^0)$

  Where x^0, y^0 are actual inputs to the system and A_1, A_2, B_1, B_2 are fuzzy sets and α_1, α_2 are firing levels. Inputs are now fuzzified and embedded in the firings of each rule.

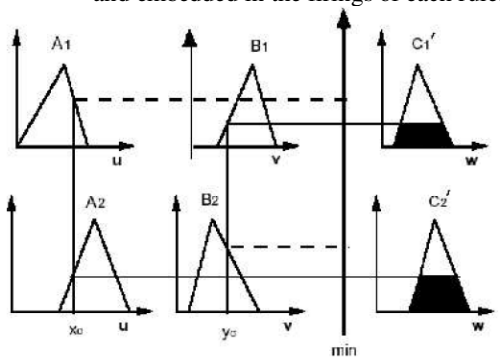


Figure 2: Mamdani Inference Mechanism

- Apply Mamdani inference mechanism [3] to get individual out fuzzy sets for each rule and a final consequence set as shown in the figure 2. Mathematically,

  $C'_1 = \alpha_1 \cap C_1$
  $C'_2 = \alpha_2 \cap C_2$

  Where C'_1 , C'_2 are individual output fuzzy sets for each rule and C is the final consequence set.

5.2 Membership Functions

In order to build membership functions (MF) it is needed to identify the ranges where each base variable is spread out.

Waiting time = distance to request + no. of stops x3
Riding time=distance to destination + no. of stops x 3

Since waiting time and riding time is evaluated with above mentioned formula range of the membership function depend on the maximum number of floors in the building. As in this system there are 15 floors maximum number of distance to arrival/destination is 14 floors. Also maximum number of stops is 14. So the range for waiting time or riding time will be from 0 to 56.

Their membership functions for waiting time and riding time are shown in figure 3a and 3b.

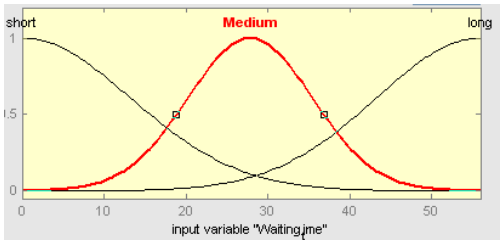


Figure 3a: Membership functions for waiting time.

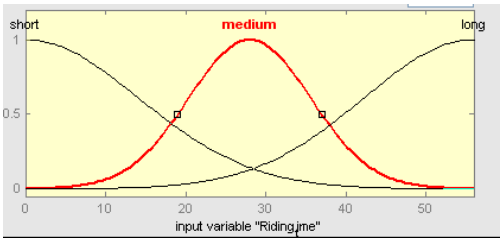


Figure 3b: Membership functions for riding time.

As maximum number of passenger for the lift is 20, the range for number of passengers is from 0 to 20. Figure 3c shows it's membership functions.

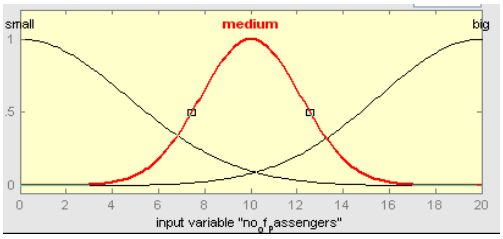


Figure 3c: Membership functions for load.

Destination call area weight and hall call area weight values represent its direction as well. So the range consists of -14 to 14. Figure 3d displays its membership functions.

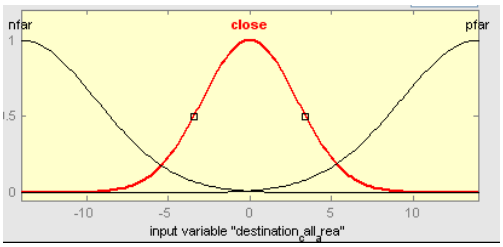


Figure 3d: Membership functions for call areas.

Membership functions small, medium and high for output 'priority' are displayed in figure 3e.

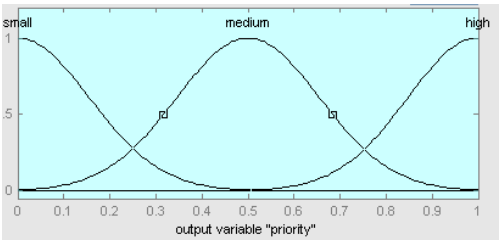


Figure 3e: Membership functions for priority.

5.3 Rule Base

Depending on the fuzzy inputs and the rule bases, the output fuzzy set, `priority' is computed using an inference scheme. Several inference schemes are available like Mamdani, Sugeno etc. For the present simulator, the Mamdani scheme has been adopted.

In this application each rule has a single input mapped to a single output to avoid complexities involved by considering all the inputs in a single rule. There are different rules to achieve different objectives as described below.

1. If waiting time short then priority is high.
2. If waiting time is medium then priority is medium.
3. If waiting time is long then priority is small.
4. If riding time short then priority is high.
5. If riding time is medium then priority is medium.
6. If riding time is long then priority is small.
7. If loading is small then priority is high.
8. If loading is medium then priority is medium.
9. If loading is high priority is small.
10. If hall call area is close priority is high.
11. If hall call area is positively far priority is medium.
12. If hall call area s negatively far priority is small.
13. If destination call area is close priority is high.
14. If destination call area is positively far priority is medium.
15. If destination call area is negatively far priority is small

Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Generally this weight is 1 and so it has no effect at all on the implication process. From time to time system may want to weight one rule relative to the others by changing its weight value to something other than 1.

5.4 Defuzzification

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set. In this system centroid method is being used.

The centroid of a plane figure can be given as the integral,

$$\frac{\int x f(x)\, dx}{\int f(x)\, dx}$$

where f(x) is the vertical extent of the object at abscissa x. This formula can be derived from the first moment about the y-axis of the area.

6. Implementation

Simulator is developed using C#.Net and MATLAB. C#.Net is used to create the graphical user interface and calculate performance of the algorithm compared to conventional algorithms. Also calculating input parameters to fuzzy logic is done in the .net

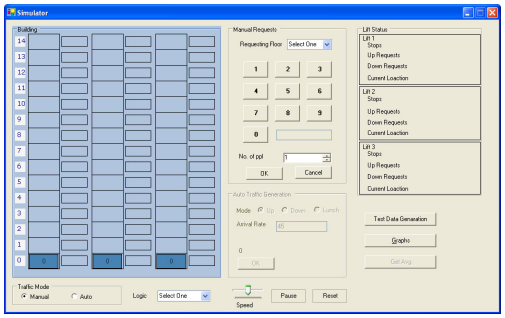


Figure 4: Snapshot of the simulator GUI

application. The dynamic link library that was created by MATLAB to evaluate the fuzzy inputs and to give the output feed the output to the .net application. In the simulator following functionalities are implemented:

- Auto traffic generator
- Simulation of few different algorithms
- Graph controller for test result analysis
- Elevator movement visualization

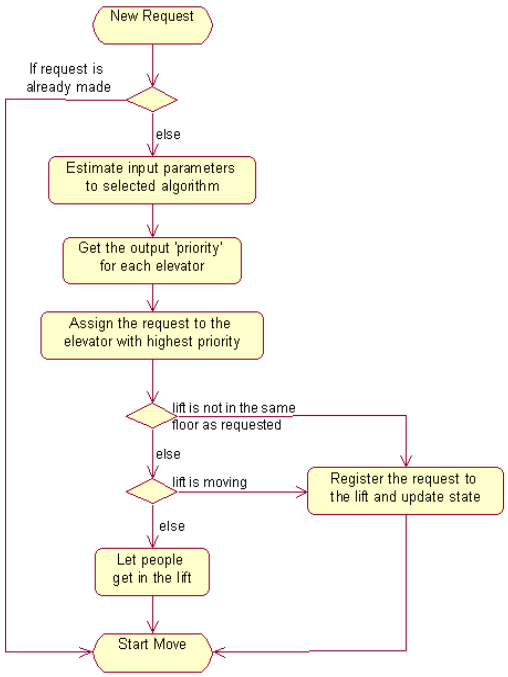


Figure 5: Request registering events flow

The above figure describes how the lift maintains its states and functions accordingly. When a new request is made system checks whether the request already exists or not. If not system estimates the input parameters to the relevant algorithms considering each lift's status. Then considering these values it estimates the priorities. If the lift with the highest priority is not in the same floor as requested or moving, it registers the request to the lift and start moving. If the lift is in the requested floor and stationary it allows people to get in and then starts moving.

7. Testing and Evaluation

In the simulation average waiting time, average riding time and average distance is mapped to graphs in order to evaluate algorithms by comparison. Figure 6a, 6b and 6c show the comparison with regard to each performance criteria. In order to have smooth line graph simulation should be done several times for each arrival rate (number of requests/ 5 minutes). In these results simulation has run 2-3 cycles for the smoothing of the results. For every graph x axis is arrival rate (number of requests per 5 minutes time period) and y axis will be the performance criteria i.e. average waiting time, average riding time, etc.

7.1 Average Waiting Time

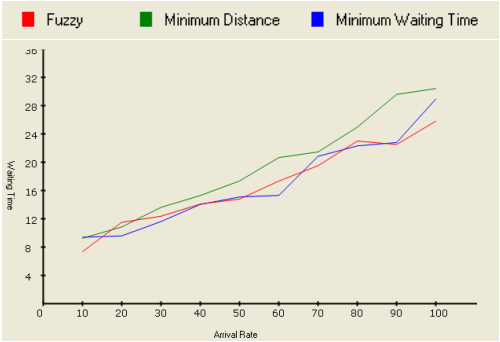


Figure 6a: Average waiting times in up peak traffic mode

Since the minimum waiting time method optimize the average waiting times it is visible that minimum waiting time method has the best performance when considering waiting times. Since fuzzy logic uses waiting time as an input to the system, it also minimize the waiting time as possible. Since minimum distance consider only the distance not the stops between the current location of the lift and the destination, its waiting times are quite high compared to other two approaches.

7.2 Average Riding Time

Fuzzy logic has used riding time as an input to the system. Figure 6b shows that fuzzy approach is being fairly good when compared to other approaches. Results are not that convenient due to that the traffic is randomly generating and simulation ran only couple of time for each arrival rate since it is time consuming.

Figure 6b: Average riding times in lunch traffic mode

7.3 Average Distance

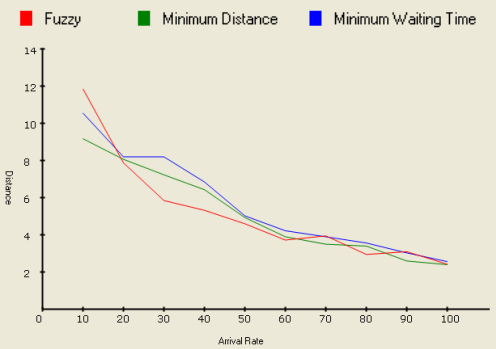


Figure 6c: Average distances in down peak traffic mode

Fuzzy logic is proven to be good in average distances as well. Around 30 – 70 arrival rates fuzzy approach is the best approach in all the traffic modes when compared to others.

When comparing all graphs fuzzy logic has optimized multiple targets very effectively unlike other approaches when one performance is optimized other performance act badly. Since in this project weight of the fuzzy rules and rules itself has been fixed, there might be some performance lost regarding some parameters. But as a future improvement if this system is enabled with dynamic rule allocation and weight changes according to the traffic mode this will optimize the needed performance at needed time.

8. Conclusion

Fuzzy logic has been proven to fit in to most of the complex situations in decision making. Elevator group controlling is one such situation where we can use fuzzy logic, i.e. rule based approaches, very efficiently. This algorithm proves to be a very good option when talking about multiple target optimizations since in new generation elevators, waiting time or riding time optimization may not be the first objective since elevator systems in high rising buildings have developed mechanically to increase elevator speeds. Power consumption may be one key thing that will look into when developing this type of system. Also fuzzy logic has shown to be quite powerful, yet computationally less intensive technique to deal with in this kind of problems.

Although destination control systems have not been installed in old buildings, it is proven to be far more effective to have destination information before entering to an elevator. With the new systems, riders can log in their destination at central lobby and the guidance programs use this information to determine which cars and how many of them will travel to specific floors. An LCD display directs riders to the proper elevator. Without traffic guidance systems, tall buildings are vulnerable to queuing, a condition that produces frustration among riders while they wait for available cars and after they jam into one, because they come into cars at random, riders end up stopping at practically every floor. And that means it's going to take a long time to come back to the lobby to pick up the next group of people which leads to more queues. But since in this system people with similar destinations ride in the same car, the number of stops the elevator has to make decreases significantly.

8.1 Future Work

This project addressed broad issues related to elevator control and was more general in nature since elevator controlling has many areas such as planning, group controlling, traffic pattern recognition, etc. Hence, there remains significant scope of improvement in most of the aspects which requires much more focused and detailed work. Some of them are listed below:

- Adaptive and learning approaches may be included to system for traffic pattern identification and parking of the elevators.
- Checking the effect of the new request assignment to the old requests and assigning lifts with minimum effect.
- Implementing fuzzy approach of group controlling in a fuzzy embedded microcontroller for the validation of the algorithm.

9. Acknowledgments

We are greatly thankful Dr. Lalith Premarathna for taking keen interest in this research and for giving valuable inputs and encouragement. Finally we are thankful to all our fellow researchers at University of Colombo School of Computing for their constant support.

References

[1] Tan Kok Khiang , Marzuki Khalid, and Rubiyah Yusof - *Intelligent Elevator control by ordinal structure Fuzzy logic Algorithm* ICARCV 97, Singapore, Dec. 1996.

[2] Marja-Liisa Siikonen - *Elevator Group Control with Artificial Intelligence*. Technical Report, Systems Analysis Laboratory Research Reports, A67, Helsinki University of Technology, 1997.

[3] Fuzzy sets and Fuzzy Logic - Theory and Applications by George J. Klir and Bo Yuan - ISBN 81-203-1136-1

[4] An Introduction to Fuzzy sets - Analyse & Design by Witold Pedrycz and Fernando Gomide - ISBN 81-203-2465-x

[5] Marja-Liisa Siikonen - *Planning and Control Models for Elevators in High-Rise Building*. Research Report A68. System Analysis of Technology, Helsinki University of Technology. 1997.

[6]http://en.wikipedia.org/wiki/Fuzzy_control_system

[7] Matthew Brand Daniel and Nikovski *Optimal Parking in Group Elevator Control* IEEE International Conference on Robotics & Automation (ICRA)

[8] Robert H. Crites and Andrew G.Barto *Improving Elevator Performance using Reinforcement Learning* Adv. Neural Inform. Process. Systems 8, 1996