SRI LANKA ASSOCIATION FOR ARTIFICIAL INTELLIGENCE

Proceedings of the 9th Annual Sessions

18th December 2012

Sri Lanka Association for Artificial Intelligence – 2012

Proceedings of the 9th Annual Sessions
18th December 2012
Nawala
Sri Lanka

Editors:

Prof. Nalin Wickramarachchi
Dr. (Ms) HUW Ratnayake

Sri Lanka Association for Artificial Intelligence, Department of Electrical & Computer Engineering, The Open University of Sri Lanka, Nawala, Nugegoda, Sri Lanka

CONTENTS

Multi-agent System Technology for Morphological Analysis

B. Hettige[1], A. S. Karunananda[2], G. Rzevski[3]
[123]A Faculty of Information Technology, University of Moratuwa, Sri Lanka,
[1]budditha@yahoo.com, [2]asoka@itfac.mrt.ac.lk, [3]rzevski@gmail.com.

Abstract-**Machine Translation involves multiple phases including morphological, syntax and semantic analysis of source and target languages. Despite there are numerous approaches to machine translations, handling of semantics has been an unsolved research challenge. We have been researching to exploit power of multi-agent Systems technology for machine translation by extending our rule-based machine translation system, BEES. Since there are no agent development framework specific to machine translation, our project has started by developing our own framework, MaSMT. This paper presents our research on the development of morphological analysis phase in MaSMT. Twenty-two ordinary agents and one manager agent have been implemented to model morphological analysis of English language. In contrast, MaSMT implements 206 agents and a manager agent to handle morphological analysis in Sinhala language. MaSMT has been developed in Java, while BEES is a Prolog implementation. Performance of morphological analysis by MaSMT and BEES has been evaluated. It was revealed that MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length such as 15 words. In case of Sinhala language too, MaSMT performs better than BEES. The difference in performances of MaSMT in Sinhala and English reflects the number of morphological rules in two languages. Due to parallel execution, MaSMT shows a significant improvement in identification of syntactic categories of words that have more than one interpretation. This feature will be reflected even better in syntactic and semantic analysis as they necessarily involve rules with multiple interpretations.**

1. Introduction

Machine Translation system is acomputer software that translates text or voice from one natural language into another with or without human assistance. In general, a machine translation system goes through several major steps including analysis of source language text, translation of syntactic categories of source into target language, and generation of the texts in the target language. Both the source language and target language sub systems are required to handle morphological, syntax and semantic aspects of the two languages. Knowledge required for language processing in those aspects can be represented as rules in the respective languages. As such, most of machine translation approaches are primarily based on rules. It is a known fact that for any given language it requires a considerable amount of rules for manipulation at its morphological, syntactic and semantic levels. For instance, we can identify almost 25 different rules to handle morphological analysis of words in English language, while Sinhala language uses 103 rules for handling morphological aspects of Sinhala words [1]. In this sense, morphological processing of a word requires to fire all such rules one after the other. Such a process is not only time consuming, but also leading to poor semantic interpretation upon the receipt of the first solution, which may not be the best. These issues have been encountered in many machine translation systems including the BEES project [2].

This research aims to improve the BEES by the existing features of multi-agent System (MAS) technology. More importantly, multi-agent system exploits the use of power of message passing among the agents that run in parallel to discover high quality solutions beyond the individual capacity of agents. We postulate that MAS technology can be used to model morphological, syntactic and semantic phases of a machine translation system.

In this paper, we present our research on the use of Multi-agent System technology for handling morphological aspects in English to Sinhala machine translation system. Despite there are hundreds of general purpose toolkits for the development of MAS, none of these have specialized for the domain of machine translation. Therefore, we have used our own MAS development environment, MaSMT, to develop the said machine translation solution. The English Morphological analyzer has been implemented with 22 ordinary agents and a manager agent to represent English morphological rules. In contrast, Sinhala Morphological analyzer has been developed with 206 ordinary agents a manager agent to represent Morphological rules in Sinhala language.

The rest of the paper is organized as follows. The section 2 describes some existing approaches for Morphological analysis. Section 3 reports multi-agent system technology for machine translation. Then section 4 briefly explains our novel approach for machine translation. Section 5 gives design of the Machine Translation system including brief description of each module of the English Morphological analyzer. After that, section 6 discusses how multi-agent based English Morphological analyzer works for the given set of words. The section 7 gives evaluation result and section 8 reports conclusion and further works of the project.

2. Approaches to Morphological Analysis

Morphology is the identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as words, affixes, parts of speech etc. [5]
The morphological analyzer reads a word as an input and identifies the stems and affixes [6]. Then it returns complete morphological information about the given word. The term "Morphological analysis" in a language has a long history. The ancient Indian linguist Panini, formulated 3,959 rules of Sanskrit morphology. Historically this is regarded as the first attempt made for the morphological analysis recorded in the world. By using the Panini Sanskrit grammar Akshar and others have developed a Panini grammar model [7,8] for all Indian language families including, Hindi, Pali, Sanskrit, etc [8]. Although the number of researchers have already used this Panini grammar model to develop morphological analyzers for their language analysis.
Anusaaraka system has developed morphological analyzers for six Indian languages [9]. Anusaaraka has been designed to translate among major Indian languages and its morphological analysis is based on the paradigms of the Indian languages. The Paradigm is used both for word analysis as well as word generation. Also AksharBharati and others have already developed a Generic Morphological Analysis Shell that can be used to develop morphological analyzers for different minority languages [10]. This Shell uses finite state transducers (FST) with features to give the analysis of a given word. The generic Morphological Analysis Shell uses dictionaries, paradigm table and paradigm classes for its Morphological analysis.
Goyal and Lehal have already developed Morphological analyzer and generator for Hindi [11]. This Morphological analyzer has been developed through the paradigm approach and implemented with Windows based GUI. This project has been developed as part of the development of a machine translation system from Hindi to Punjabi Language.
Morphological analyzers for English language have been developed by many researchers. Among others, Koskenniemi's two-level morphology [12] was the first practical and most general model in the history of computational linguistics for the analysis of morphologically complex languages. Koskenniemi's Pascal implementation of morphological analysis was quickly followed by others. The most influential of them was the KIMMO system by LauriKarttunen [13] and his students at the University of Texas. PC-KIMMO is yet another morphological analysis tool, which was based on Koskenniemi's work and implemented in C. Among others, PC-KIMMO is supposed to be the only available free English morphological analyser with a wide coverage [14]. The lexicon used in PC-KIMMO considers verb, pronoun, noun, prepositions, adverbs and adjectives. The current version PC-KIMMO is implemented in C and can be run on a PC [15]. The PC-KIMMO accepts an input word from a user, and provides all possible morphological details of the word.

3. Multi-agent Systems Technology for Machine Translation

The multi-agent system technology is a modern approach for machine translation which is used to handle complex knowledge. In general multi-agent system contains four key components namely Multi-Agent Engine, Virtual world, Ontology and Interfaces [18]. The Multi-agent engine provides a run time support for agents. Virtual world is the environment of the multi-agent systems. The Ontology contains conceptual problem domain knowledge of each agent.

A. Existing MAS Development for Natural Language Processing

Considering the existing Natural Language Processing (NLP) approaches only few Multi-agent systems are available. Minakow and others [3] have developed a Multi-agentbased text understanding system for car insurance domain. This system uses Multi-agent system based approach to understand a given text. The system uses four steps to text understanding namely morphological analysis, syntax analysis, semantic analysis and pragmatic analysis. To analyze, the whole text is divided into sentences. Then first three stages are applied to each sentence. After analysing each paragraph, text is passed to pragmatic analysis.
Stefanini and others [4] have also developed a Multi-agent based general Natural language processing system named Talisman. The Talisman agents can communicate with each other without the central control. These agents are capable to directly exchange information using an interaction language. Linguistic agents are governed by a set of local rules. The TALISMAN deals with ambiguities and provides a distributed algorithm for conflict resolutions arising from uncertain information

B. Existing MAS Development frameworks

Frameworks save developer time and also aid in the standardization of Multi-agent System development. There are number of standard frameworks available for multi-agent system development including JADE, AgentBuilder, SeSAm etc.

JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language [19]. JADE is a middle-ware that complies with the FIPA specifications. This framework provides supporting GUI tools for debugging and deployment phases in the multi-agent development. In addition to the above, agent platform can be distributed across machines and the configuration can be controlled via a remote GUI [20]. JADE successfully work with JRE 1.4 or above.

AgentBuilder [21] is an integrated software development tool that allows software developers with no intelligent agent technologies to quickly and easily build intelligent agent-based applications. AgentBuilder consists with three versions such as Lite, Pro and Pacs. AgentBuilder Lite provides several tools for development including Project Manager and Ontology Manager. The Ontology Manager provides tools for creating ontologies and automatic code generation using graphical object modelling tools. AgentBuilder Lite distributions are available for the Windows and Linux Platform.

SeSAm [22] (Shell for Simulated Agent Systems) provides a generic environment for modelling and experimenting with agent-based simulation. SeSAm provides tool for construction of complex models easily. SeSAm consist with several features and tools including visual agent modeling, Flexible environment and simulation analysis.

Agent based solution of the BEES project has been already developed to translate English text into Sinhala through the 9 agents [23][24]. These agents use existing rule-based translation module available in the BEES including English Morphological analyzer, English parser, Translator, Sinhala Morphological generator and Sinhala Sentence composer. As a result of that, core of the analysis is also rule-based in this previous system

4. A Novel Approach to Machine Translation

The approach behind MaSMT is based on the hypothesis that words employ as the building block of natural language understanding. This is valid for people who read a sentence word by word or otherwise by locating selected words such as nouns and verbs. Consequently, meaning of a sentence is determined by the interaction among words, which draw from all aspects of morphology, syntax and semantic, as appropriate. Thus MaSMT define words in a sentence as the agents. The agents pass messages among them within and across different level of analysis. As such, our machine translation approach is different from existing ones that sequentially define linguistic aspects such as morphological, syntax and semantics analysis.

For example, assume that we change our original understanding of a sentence when we read, say, from 5th word to the 8th word in the sentence. In such situations, we might continue to operate at the syntax level analysis or proceed to seek semantic information before completing the sentence at the syntax level. Thus syntax and semantic processing are inter-wound or parallel, not necessarily sequential. We argue that this connotation in language processing is much closer to how human beings process natural languages, and this can be effectively implemented by Agent technology

5. Design

MaSMT (Multi-agent System for Machine Translation) is a Java-based multi-agent development framework that can be used to develop Machine Translation applications. MaSMT provides two types of agents as ordinary agents and manager agents. A manager agent consists with number of ordinary agents within its control. Further, manager agents can directly communicate with other manager agents and the ordinary agents in the swarm that is assigned to a particular manager agent. Agents in a swarm can directly communicate only with the agents in the own swarm and the relevant manager agent. The framework primarily implements object-object communication, managing the agents (e.g. creation and execution), XML-based data passing and MySQL database connectivity for manipulation and use of the domain ontology. Agent communication in the framework has been implemented to comply with FIFA-ACL specifications [25].

This framework also provides MySQL database connector, XML database connector, message viewer and the agent monitor. Two connectors are used to communicate with MySQL database or XML database. A message viewer tool is used to view each message in the given message queue. This tool has been used to show message passing process in the MaSMT framework. The agent monitor is used to show each agent state (active, working, dead or busy).

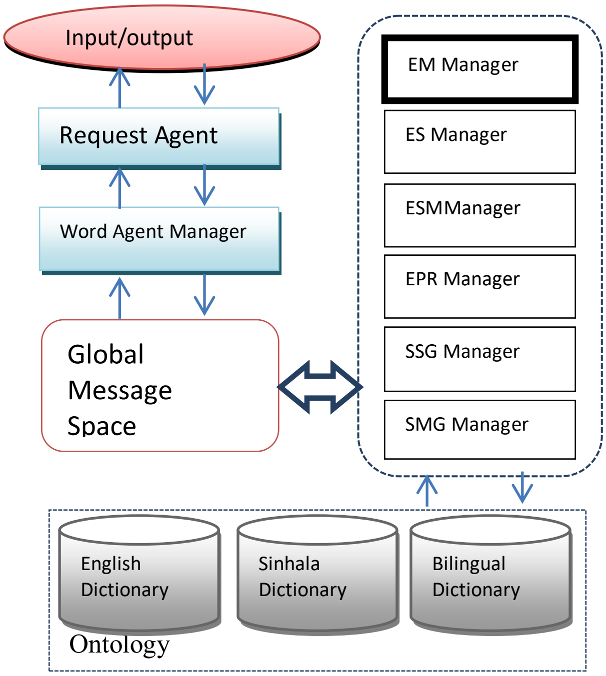

Figure1: Top level design of the Multi-agent based English to Sinhala machine translation system.

Through the MaSMT framework English to Sinhala Machine Translation system has been designed with 10 modules. Namely 7 managers, request agent, message space and the ontology. Word agent manager, English Morphological manager (EM), English Syntax manager (ES), English Semantic manager (ESM), English pragmatic manager (EPR), Sinhala Syntax manager (SS) and Sinhala Morphological manager(SM) are the 7 managers of the system. Three lexical dictionaries namely, English dictionary, Sinhala dictionary, English-Sinhala Bilingual dictionary is used as ontology. Figure 1 shows top level design of the English to Sinhala machine translation system.

System reads English text as an input and provides translated Sinhala text as an output text. The request agent makes a request to translate the given text. Then word agent manager assign agents for each words in the given text. Therefore a word in the text is worked as an agent [23]. These word agents can communicate with each other agent(s). Further, message communication architecture has been designed through the MaSMT framework. The English Morphological manager controls all the functionalities on the English Morphological analysis. This Morphological processing system consists with 5 major components namely global message space, Morphological manager, Morphological agents, local message space and ontology. Figure 2 shows the design diagram of the Morphological processing system.

A. Morphological Manager

Morphological Manager manages its client agents. According to the MaSMT framework, each manager can fully control its client agents. Therefore, manager can create, remove or control its client agent(s) in the group. Manager agent creates all its clients automatically at the initialization stage. Morphological manager reads morphological rules which are available in the rule-base (part of the ontology) and assign each rule for a client agent. For instance, 85 grammar rules are available for Sinhala noun. Therefore 85 ordinary agents are created to implement the grammar). In addition to the above, manager can directly access each agent and send messages directly for its clients. The Manager agent reads input massages from the global message queue and provides relevant tasks for the client agents. Also Manager can control the priority of the agent and the stage of the clients. This facility removes the unnecessary work load from its client agents.

B. Morphological Agents

Morphological agents work under the control of the manager agent and each Morphological agent must have a manager agent. Morphological agent is a simple java based program (Thread) which support to do limited task(s). These Morphological agents can communicate with each other through the messages space and use peer-to-peer communication method. Morphological agent contains local message queue, Morphological rules and the ontology. Agents are responded for the messages which are available in its message queue. Each agent has been assigned for only the simple task and it response only for the assign task. Morphological agents response only two messages which contain “how are you” and “who am I”. The agent reserved message “who are you” from the message queue then agent provides information about itself to the message sender by using message space. After reserving the message “who am I”, it tries to do the Morphological analysis with support of the rules and its ontology.

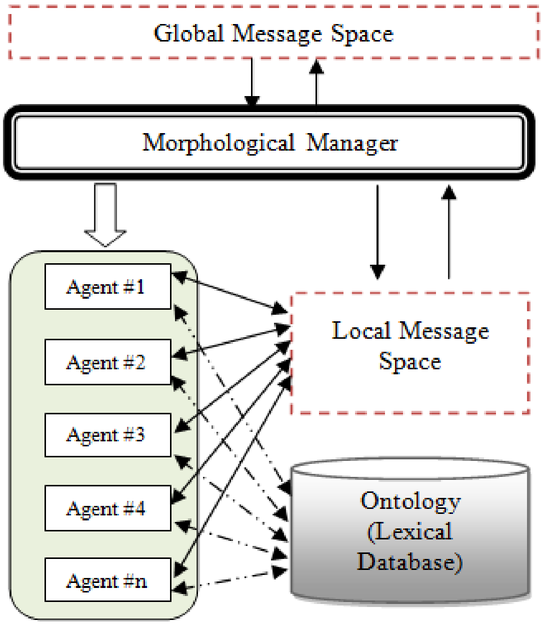

Figure2: design of the Morphological processing system

C. Local Message Space

Local message space is the message space for the each local agent group (Swarm). Each agent can directly communicate with each other through the local message space. This allows peer-to-peer connection. Morphological agent can send or receive messages from its local message space.

D. Global Message space

Global message space is the message space which can be used to access the system. This message space is visible only for the managers who are in the MaSMT framework. This message space uses for communication between managers. Agents can send messages only for the local messages and its managers.

E. Ontology

Ontology is the knowledge of each agent. English dictionary and the Sinhala dictionary are work as the ontology of the Morphological processing system. The English dictionary has been stored as a MySQL database. Further, 8 tables are used to store relevant lexical information for the English dictionary such as eng_reg_noun, eng_reg_verb, eng_reg_adjt, eng_irr_noun, eng_irr_verb, eng_irr_adjt, eng_irr_words and eng_pro_noun. To store morphological information system uses 2 tables namely eng_noun_mop_ruleandeng_verb_mop_rule.

Similarly there are another 8 tables to store relevant lexical information for the Sinhala dictionary such as sin_reg_noun, sin_reg_verb, sin_reg_adjt, sin_irr_noun, sin_irr_verb, sin_reg_advb, sin_reg_prep and sin_reg_conj. To store morphological information system uses 3 tables namely sin_noun_mop_rule, sin_verb_mop_rule and sin_noun_case_rule These two dictionaries have been developed based on the existing prolog dictionaries available in the BEES project[26].

F. Messages

Messages have been used to communicate with each other. These messages are developed using FIPA ACL message stranded. ACL Message consists with Participant in communication: sender, receiver, reply-to, Content of message: content, Description of Content: language, encoding, ontology. Control of conversation: protocol, conversation-id, reply-with, in-reply-to, reply-by etc.

Using the above design structure English and Sinhala morphological analyzers have been developed. The English Morphological analyzer has been implemented 22 ordinary agents and a manager agent for modeling morphological rules in English language. The English Morphological analyzer uses English dictionary as its Ontology. The English dictionary consists of more than 35000 English words including more than 20000 regular and irregular nouns, more than 10000 verbs and more than 5000 adjectives. In contrast, Sinhala morphological analyzer has been implemented 220 agents and a manager agent with regard to handling of nouns and verbs morphology in Sinhala language. The Sinhala Morphological analyzer uses Sinhala dictionary as its Ontology. The Sinhala dictionary consists with more than 80000 Sinhala words including 45000 nouns, 15000 verbs and 15000 other words.

6. How English Morphological Analyzer Works

This section describes how multi-agent based morphological analyzer works for a given English text. Figure 3 shows user interface of the English Morphological analyzer and figure 4 shows user interface of the Sinhala Morphological analyzer.

As a first step, the request agent makes a request to analysis. Then word agent manager reads the input text and word agents are automatically created for each word in the text. After that, each word agent ask message from English morphological agent "who am I". Word agent manager receives these messages from the word agents and send it to the English morphological manager through the global message space. The English morphological manager receives these messages and sends to its clients. (At this point each morphological agent has messages on their local message queue). The English morphological agents read these messages with the title "who am I" and try to analyze it with existing morphological rule. If rule is accepted then agents send relevant grammar information for the message sender through the English morphological manager. The English morphological manager sends these reply messages for the English word agent manager to deliver to its clients.

The Sinhala Morphological analyzer works with the same architecture of the English Morphological analyzer. The Sinhala Morphological agent also response the messages "how am I" and do the Sinhala morphological analysis.

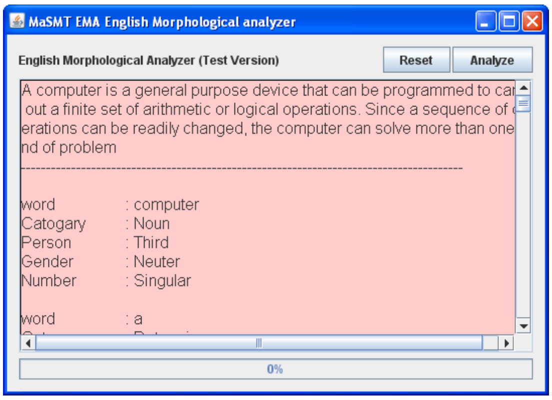

Figure3: User interface of the English Morphological analyzer

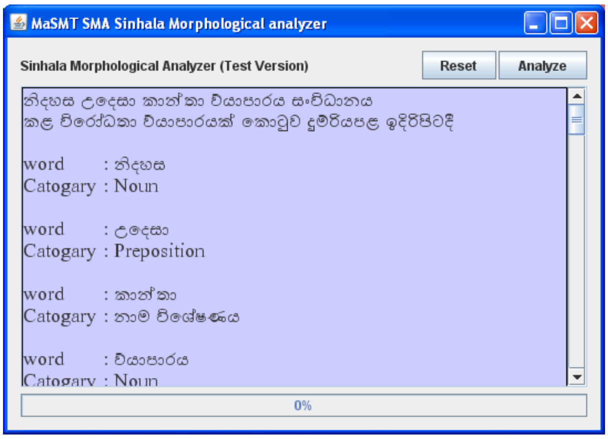

Figure4: User interface of the English Morphological analyzer

7. Evaluation

Two multi-agent based Morphological analyzers have been evaluated separatelythrough the existing evaluation methods which was used in the rule-based morphological analysis under the BEES project [27]. The English Morphological analyzer has been tested through the created test plan including 50 test cases and 100 sample words. The Sinhala morphological generator has been evaluated through the same evaluation method which is used to evaluate English Morphological analyzer. The Sinhala Morphological generator has been tested through the created test plan including 150 test cases and 300 sample words. Table 1 shows the evaluation results of the English and Sinhala morphological analysis

Further, two analyzers also tested against the rule-based morphological analyzer in BEES [16]. Figure 5 shows the experimental result of the rule based and Multi-agent based Morphological analysis. In the experiment we have separately calculated the time taken to analyze words in rule-base and multi-agent systems. The experimental result shows that, MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length such as 15 words. In case of Sinhala language too, MaSMT performs better than BEES. The difference in performances of MaSMT in Sinhala and English reflects the number of morphological rules in two languages.

Table 1: Evaluation Results of the English Morphological Analysis

Criteria	English Morphological Analyzer	Sinhala Morphological Analyzer
Success	92	260
Over-specified	-	-
Under-specified	5	28
Wrong Analysis	-	-
Over &under-specified	-	-
Irrelevant		
Not found	3	12
Total Solutions	100	300

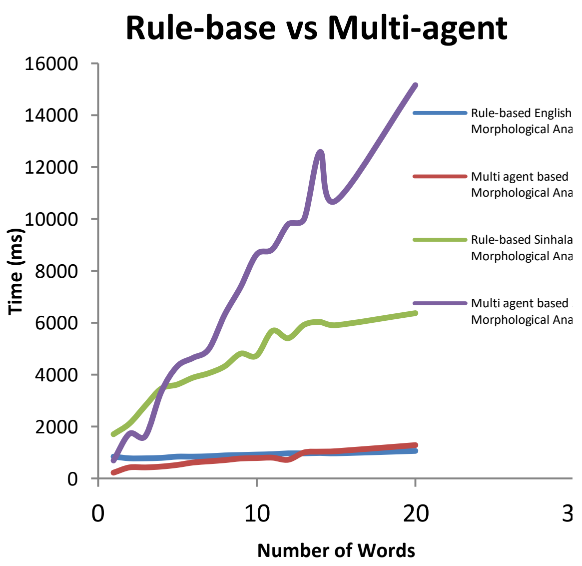

Figure5: performance comparison between rule-based vs. multi-agent approaches

8. Conclusion and Further works

This paper has reported our research on the use of MAS technology for handling morphological aspects in English to Sinhala machine translation. English and Sinhala morphological analyzers have been developed through our own MaSMT framework. MaSMT implements 22 ordinary agents and a manager agent for modeling morphological rules in English language. In contrast, MaSMT implements 206 agents and a manager agent with regard to handling of nouns and verbs morphology in Sinhala language. Two morphological analyzers have been tested through the created test plan. The experimental result shows 97% accuracy of the English morphological analysis and Sinhala morphological analyzer shows 96% accuracy. In addition to the above, both morphological analyzers have been compared with rule-based implementation in BEES. Experimental result shows that MaSMT performs much faster than BEES for morphological analysis of English sentences with a reasonable length

Due to parallel execution, MaSMT shows a significant improvement in identification of syntactic categories of words that have more than one interpretation. This feature will be reflected even better in syntactic and semantic analysis as they necessarily involves rules with multiple interpretations

Use of Multi-agent Systems technology to implement the other phases, namely, syntax and semantics analysis in machine translations has been considered as the key further work of MaSMT.

References

[1] B. Hettige, A. S. Karunananda, "Computational Model of Grammar for English to Sinhala Machine Translation", Proceedings of the International Conference on Advances in ICT for Emerging Regions - ICTer2011, Colombo, 2011.

[2] B. Hettige, (2011) BEES: Bilingual Expert for English to Sinhala, [online]. Available: http://dscs.sjp.ac.lk/~buddhitha/bees.htm

[3] I. Minakov and et al., “Creating Contract Templates for Car Insurance Using Multi-Agent Based Text Understanding and Clustering”, Third International Conference on Industrial Applications of Holonic and Multi-Agent Systems., HoloMAS:, 2007, pp 361-371.

[4] M. H. Stefanini, Y. Demazeau, “TALISMAN: A multi-agent system for natural language processing”, In Proceedings of SBIA'95. - Springer Verlag:, 1995, pp. 312-322.

[5] Wikipedia,[online], Available: http://en.wikipedia.org.

[6] D. Jurafsky, J. H. Martin, “Speech and Language Processing”, Boulder: University of Colorado, 2005.

[7] B. Akshar, V. Chaitanya, R. Sangal, “Natural Language Processing: A Paninian Perspective”, New Delhi, India, Prentice Hall of India, 1995

[8] S. Bhate, S. Kak, “Panini's Grammar and Computer Science”, Annals of the Bhandarkar Oriental Research Institute, vol. 72. 1993, pp. 79-94.

[9] Anusaaraka, [online]. Available: http://anusaaraka.iiit.ac.in/.

[10] B. Akshar, R. Sangal, D. M. Sharma,R. Mamidi, “Generic Morphological Analysis Shell”, In Proceedings of LREC, 2004.

[11] V. Goyal, G. S Lehal, Hindi Morphological Analyzer and Generator, First International Conference on Emerging Trends in Engineering and Technology, IEEE, 2008 pp 1556-1559.

[12] K. Koskenniemi, “Two-level morphology: A general computational model for word-form recognition and production Publication”, University of Helsinki, Department of General Linguistics, Helsinki, 1983.

[13] L. Karttunen, R. B. Kenneth, “Twenty-five years of finite-state morphology”, In Inquiries into Words, Constraints and Contexts. CSLI Publications, 2005, pp71-83.

[14] K. Daneilf, Y. Schabes, M. Zaidel, D. Egedi, “A Freely Available Wide Coverage Morphological Analyzer for English”, Proceedings of COLIN-92. 1992, pp. 23-28.

[15] L. Karttunen, R. B. Kenneth, “Twenty-five years of finite-state morphology”, In Inquiries into Words, Constraints and Contexts. CSLI Publications, 2005, pp71-83.

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

[17] SWI-Prolog Home Page, [online] Available:http://www.swi-prolog.org/index.html.

[18] G. Rzevski, “A new direction of research into Artificial Intelligence”, Sri Lanka Association for Artificial Intelligence 5th Annual Sessions.- 2008.

[19] JADE, [online], Available: http://jade.tilab.com/

[20] F. L. Bellifemine, G. Caire, D. Greenwood, Developing Multi-Agent Systems with JADE, John Wiley & Sons, Ltd, 2007.

[21] Agent Builder, URL: http://www.agentbuilder.com/Documentation/Lite/

[22] SeSAm, [Online], Available: http://www.simsesam.de/

[23] B. Hettige, A. S. Karunananda, "A Word as an Agent for Multi-agent based Machine Translation", Proceedings of the ITRU Research Symposium, Moratuwa, 2011

[24] B. Hettige, A. S. Karunananda, "Multi-Agent architecture for English to Sinhala Machine Translation", Proceedings of the 27th National IT conference (NITC10), Sri Lanka. 2010.

[25] FIPA:Agent Communication specifications, [online], Available: http://www.fipa.org

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

[27] B. Hettige, A. S. Karunnanda., "An Evaluation methodology for English to Sinhala machine translation", Proceedings of the 6th International conference on Information and Automation foe Sustainability (ICIAfS 2010), IEEE., 2010.

Multi-Agents to Work with Primary Child: Expert Agents Architecture and Implementation

M.S. Nanayakkara[1], K.A. Dilini T. Kulawansa[2]
[1]NAPD – OPD Saudi Aramco, WorleyParsons Arabia Ltd. Co., P.O. Box 31699 – Al Khobar 31952, Kingdom of Saudi Arabia
[2]Department of Computational Mathematics, Faculty of Information Technology,University of Moratuwa, Sri Lanka
[1]madhusha.sn@live.com,[2]dilinik@uom.lk

Abstract–**As an important part of today's socio-techno-ecosystem, it is very vital to acknowledge the importance of primary education. Problems like fair distribution of resources, resources utilization, and microscopic concern on aspects related to the psychology of a child cause any primary education system, globally, face significant challenges when being implemented.CHILD@EDU is an alias given to a piece of research and development work which has being initiated to find solutions for these matters in primary education. CHILD@EDU fundamentally targets the children that involve in Primary Education whileproposing a knowledge system that addresses the vital and sensitive attributes of a common primary education system (known as the fundamental characteristics of a child with effective learning abilities). This paper is intended to publish the technical research, architecture, and implementation work related to the Expert Agents module of CHILD@EDU. This distributed multi-agent architecture has been tested in the means of project CHILD@EDUto be functioning with a higher accuracy adhering to the ontological expert decision support based on international WISC®-IV Assessment Standards.**

1. Introduction

The time that the child spends with the teachers determines the best quality aspects of his/her entire life. As the first step of the compulsory education, it is very vital to acknowledge the importance of primary education.

However, any primary education system faces some key problems, especially, when addressing the needs of measuring children's hidden intelligence and cognitive abilities. The term "CHILD@EDU" has been used as an alias given tothe research and development work which was initiated to find solutions for these matters in primary education.[11] CHILD@EDUconsisted of two major phases to solve the addressing problems:

- Research on Current Primary Education System and Child Psychology
- The Internet based knowledge system

CHILD@EDUhas been implemented using Multi Agent Artificial Intelligence Technology with the support of other AI and software design technologies such as Ontology [6], Web Services, Web 2.0 rich web applications, and WPF network oriented user interfaces. Its physiological evaluation is based on the international standard of WISC-IV [1, 3].

This paper is intended to publish the technical research, architecture, and implantation details of CHILD@EDU Expert Agents module. In demonstration purposes, it has been developed using JADE and integrated to CHILD@EDU Java service-application-core. Nevertheless, this architecture is strongly coupled with WISC®-IV Assessment Standards Ontologies derived from WISC-IV rules. However, the objective of such exemplary implementation is to showcase the abilities that Multi-Agent technology has over real-time school education systems.

2. Problem in Brief

[11]Any primary education system, globally, faces some key problems that cannot be addressed via direct human agency or the application of any ordinary information system. Science has proved that a primary child's mental processing is very difficult to be understood by ordinary human beings due to the lack of significant and stable modes of communication that the child has with the environment. Scholars show that trying to develop a child to near perfect human being is just like trying to finish a micro sculpture with higher level of details [7]. Thus, a special framework/approach is required to address the following problems to achieve effective primary education:

- One Teacher-One Child education is not globally possible due to lack of resources and skilled personnel.
- All Teachers or parents are not expert child psychologists or physiatrists.
- Effective report generation of a child's educational attributes is important.
- Who supports the task of teachers? With the growing complexity of environment, even the teachers must be guided to be productive.

- Parents, Education Institute, and Children must be integrated.
- System of Systems is needed for efficient government decision making.

With the increasing complication of this problem domain, a consistent methodology to model its complexity is required in order to let a digital system work hand-in-hand with teachers. Thus, in order to achieve it, there should be a dependable manual standard, set of rules, or a protocol for educating, interacting with, and evaluating children as well as a technology that supports effective modeling of knowledge derived from that manual mechanism.

3. Overview- JADE and WISC-iv

JADE (Java Agent DEvelopment Framework) and Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV)have been prioritized as the key technologies of this research. The project's aim has been set to find the application of these technologies to produce a knowledge system that can support primary education to increase its quality and resource utilization.

A. JADE (Java Agent DEvelopment Framework)

JADE (Java Agent DEvelopment Framework) [9] is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent [11] systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. JADE is completely implemented in Java language and the minimal system requirement is the version 1.4 of JAVA (the run time environment or the JDK) [9, 15].

JADE heavily supports the implementation of CHILD@EDU as well. Especially due to its supportability and compatibility with Protégé, the compliance Java code can be easily generated from the ontology conceptual model [15].

B. Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV)

Despite all the innovations and exemplary quantitative and qualitative characteristics of new and recently revised intelligence tests, the Wechsler scales continue to reign supreme. In fact, the Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV) [1, 3], like its predecessor—the WISC-III—will very likely become the most widely used measure of intelligence the over world. Because the latest edition of the WISC represents the most substantial revision of any Wechsler scale to date, CHILD@EDU's task of developing an interpretive technology system for evaluating children and monitoring their learning and intelligence attributes using WISC-IV [1, 3] becomes psychometrically, technologically, and theoretically defensible.

This scientific approach for evaluating children offers 10 types of compulsory subtests and 5 optional subtests for taking measurements. The following are the 10 compulsory tests that are simulated through CHILD@EDU:

i. ***Block Design (BD):*** The examinee is required to replicate a set of modeled or printed two-dimensional geometric patterns using red and-white blocks within a specified time limit.
ii. ***Similarities (SI):*** The examinee is required to describe how two words that represent common objects or concepts are similar.
iii. ***Digit Span (DS):*** On Digit Span Forward, the examinee is required to repeat numbers verbatim as stated by the examiner. On Digit Span Backward, the examinee is required to repeat numbers in the reverse order as stated by the examiner.
iv. ***Picture Concepts:*** The examinee is required to choose one picture, from (PCn) among two or three rows of pictures presented, to form a group with a common characteristic.
v. ***Coding (CD):*** The examinee is required to copy symbols that are paired with either geometric shapes or numbers using a key within a specified time limit.
vi. ***Vocabulary (VC):*** The examinee is required to name pictures or provide definitions for words.
vii. ***Letter-Number:*** The examinee is read a number and letter sequence and is Sequencing (LN) required to recall numbers in ascending order and letters in alphabetical order.
viii. ***Matrix Reasoning:*** The examinee is required to complete the missing portion (MR) of a picture matrix by selecting one of five response options.
ix. ***Comprehension:*** The examinee is required to answer a series of questions (CO) based on his or her understanding of general principles and social situations.
x. ***Symbol Search (SS):*** The examinee is required to scan a search group and indicate the presence or absence of a target symbol(s) within a specified time limit.

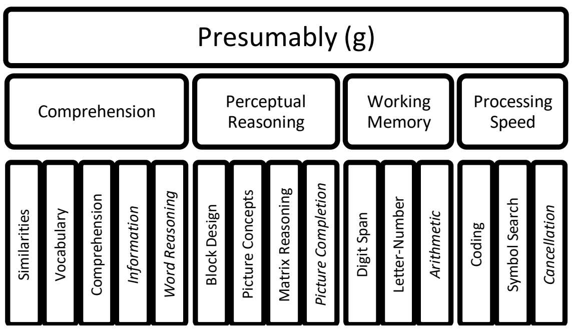

Figure1: WISC-IV Subtests for Evaluating Children

4. Overview of The Approach: CHILD@EDU and Expert Agents Module

A. Overview of CHILD@EDU Approach

As it was mentioned above, the project CHILD@EDUhad been focused to research an effective way to involve with a primary child while teaching, monitory, and evaluating him/her. Its basis of study has been influenced by characteristics of a successful child in education, measurement methods to evaluate a child, standards of Primary Education Systems, mechanism to introduce computer agency instead of human agency, and the cardinality and capacity of required Computer Agency.

The most important aspect of CHILD@EDU is that as a final output it generates activities for children which target a composition of measuring subtests. These activities will be in either one of 5 types:

i. Reading
ii. Writing
iii. Symbolic
iv. Numeric
v. Motor

However, most of the world wide syllabuses (case studied on Sri Lanka Local and Cambridge Syllabuses) group these five subject areas into three classes from Elementary to Grade three education as follows;

i. Language,
ii. Numbers and Mathematics,
iii. Environmental Studies

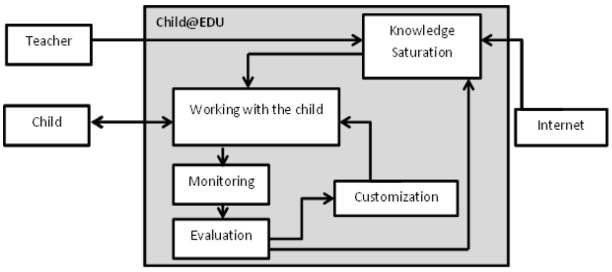

Figure2: CHILD@EDU Generic Process Flow

In abstract viewpoint, the system acts as a testing surface to measure and monitor the educational progress of a primary child. In order to support this task, international standard of Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV) [1, 3] has been used. However, the main difference between an ordinary system and CHILD@EDU is that the extra focus it gives to employment of expert software agents of WISC-IV [1, 3] that monitor, evaluate, and report on a particular micro area of the learning child. This helps the system to identify talented children as well as children with deficiencies and automate the progression of learning curve of each individual child effectively through customization of course work and deploying teamwork. The summary of the process of this CHILD@EDU prototype can be elaborated as in the above Figure.

B. Overview of CHILD@EDUExpert Agents Module

The solution for handling the complexity of the domain, working with the adhering standards of WISC-IV, and modelling the knowledge and the rules, as CHILD@EDU suggests, is the use of distributed, work-delegated, multi-swarm expert agents. These computer agents may focus on micro aspects of the primary education mechanism and the standard that the underlying Ontology and their local ontologies define. As CHILD@EDU adheres to WISC-IV international standard, its expert agents also provide services based on WISC-IV from generating activities for the children (through the provided Man-Machine interface) to monitoring and evaluating their activity.

5. Design of CHILD@EDU andIts Expert Agents Module

A. Overview of the design of CHILD@EDU

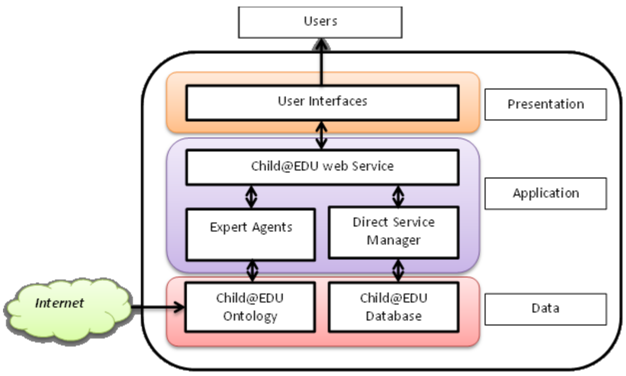

Figure3: Top Level Architecture of CHILD@EDU

Figure3 shows the top level architecture of this CHILD@EDU prototype which will be the output of this project. The architecture contains three main layers, as discussed above: Data Layer, Application Layer, and Presentation Layer.

- ***The Data/Knowledge Layer*** contains the main modules of CHILD@EDU ontology and the non-ontological relational database. The ontology [6] uses the trusted sources from the Internet in addition to the user inputs in the knowledge saturation process.
- ***Application Layer*** contains three main modules: the expert agent module, which employs multi-agent technology, direct service manager module to provide encapsulated logic of accessing the relational database to the web service, and CHILD@EDU web service itself.

- ***Presentation Layer*** contains the Network based user interface provided for the users of the system. The GUI is web oriented due to the technical feasibility and to ensure higher availability anywhere anytime.

B. Expert Agents Module

The most important as well as the complex module of CHILD@EDU is the Expert Agents Module, which resides in its application layer. It contains a pool of software agent swarms. Among the application layer agents, the tasks of creating activities for children that composes few subtest elements together and operating subtests separately for evaluation run separately. The knowledge for the expert agents to function on comes from the underlying CHILD@EDU Ontology. In abstract view, agent swarms operate in two categories as;

Activity Operating Agents:

These are the agents that construct composite activities adhering to multiple subtests or one subtest. They are responsible of;

- Producing actives adhering to the recommendations given by the experts of relevant subtest/s.
- Transferring the metadata required to generate user interfaces in presentation layer.
- Retrieve the response parameters from user-interfaces related to the users' interaction.
- Delegating these response parameter to expert agents for evaluation

There are three (3) swarms of agents of this nature in order to generate activities of five main modes

- *Language Activity Generators*
- *Numeric Activity Generators*
- *Environmental Activity Generators*

WISC-IV Oriented Subtest Expert Agents:

These agents has specialized knowledge in each subtest closely related to WISC-IV [1, 3] standard. These agents provide the insight needed for the Activity Operating Agents to produce activities and evaluate the delegated responses related to the relevant subtest. For each major subtest out of all ten (10) there exists a swarm of agents.

In addition to that, there are five significant agents:

i. ***The central message space***: Responsible for the coordination of all the other swarms
ii. ***Course conductor:*** Responsible of structuring and conducting a course according to the progression and formulate the exercises in a particular course.
iii. ***General Evaluator***: Responsible of evaluating exercises and a course as a whole. These agents are also responsible for combining the evaluations of each of the evaluations made by the other expert agents and output the progress measurements of the child on that particular exercise and in a time series manner.
iv. ***Class Administrator***: Responsible of directing the class with respect to managing the student profiles, teacher profiles, and other class infrastructure oriented tasks of a class
v. ***Web Service Agent***: A new concept, to support the concurrency model of web service. This agent allows the Agent Experts Module to be loosely coupled with the Web Service module and acts as the Only Request Agent for the central Message Space.

Each of these swarms follows the traditional architecture of Request-Resource-Message-Ontology multi-agents [12]. The following figures demonstrates the adoption of this architecture and the way the sample swarm communicates with the central message space agents.

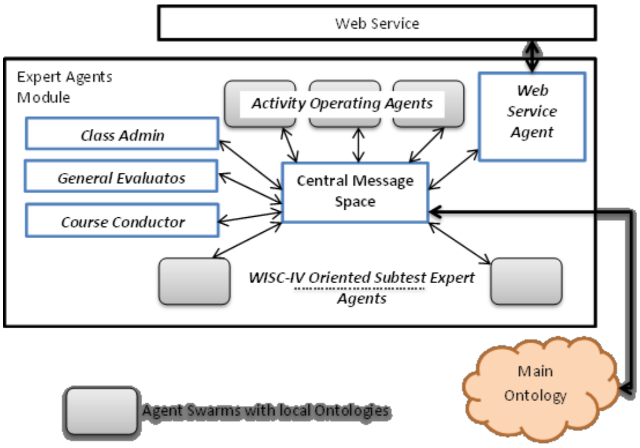

Figure4: Top Level Design of Expert Agents Module

6. Implementation Summary of CHILD@EDU Expert Agents Module

As it was clearly shown in last chapter, CHILD@EDU occupies a pool of software agents as swarms to achieve higher efficiency in software process and knowledge handling. As it was discussed there are two types of agents as Activity Operating Agents and Subtest Expert Agents and three other uncommon swarms of agents as Course Conductors, General Evaluators, and Central Message Space.

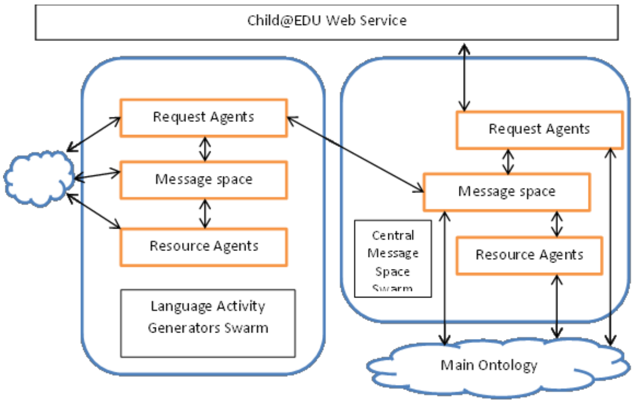

Figure5: Multi-Agent Architecture of Expert Agents Module.

As the key technology of implementing agents, JADE plays a crucial role [9]. From initiation though execution, suspension, and resumption to the termination of agents. JADE is a middleware that facilitates the development of multi-agent systems acting as a platform for Agents and a container [12]. It includes;

i. A runtime environment where JADE agents can "live" and that must be active on a given host before one or more agents can be executed on that host.
ii. A library of classes that programmers have to/can use (directly or by specializing them) to develop their agents.
iii. A suite of graphical tools that allows administrating and monitoring the activity of running agents.

Thus, the developer only has to worry about designing the agent swarms, defining their behaviours, defining their communication, and linking them to the knowledge sources.

This prototype of CHILD@EDU employs 16 different types of agent swarms to reach for its heights. These include five (5) different Activity Operator swarms, ten (10) Subtest Expert swarms for each subtest with relevance to WISC-IV [1, 3], and three special swarms to maintain the consistency of the system.

A. Agents in an Activity Operators Swarm

As it was mentioned earlier, there are three (3) different activity operator agent swarms for generating writing and alphabet reacted activities, Symbols related activities, numeric and mathematics activities, and creativity and motor aspects related activities of the following syllabus.

The agents in these swarms have fewer similarities with contrast to the agents in Subtest Swarms. The behaviourof these agents generally are for formulating and carrying out a learning or testing activity with the child. Though they are dependent on the data they receive via the web service, they consider that the responses directly come from the users of the system. The agent of an Activity Operating Swarms can be categorized into two classes.

i. Activity Generators: Responsible for generating a learning or testing activity for the child from a collaborative effort.
ii. Activity Evaluators: Deal with the responses of the user and conclude the activity

The following flow of steps explains the abstract functionality of and Activity Operating Agent Swarm.

i. The Request Agent (RA) of the relevant Activity Operating Swarm is spawned by the Central Message Space (CMS)
ii. CMS provides the RA the information on what type and level of activities must be generated for the active child
iii. RA, though the local message space of the swarm spawns the activity generators (AGs)
iv. AG agents, though the central RA and Central Message Space call upon the service of relevant Subset Experts for generating the Activity
v. AGs generates the activity once the Subtest knowledge is received
vi. AGs passes the activity meta-knowledge to RA which is transferred to the presentation layer through web service
vii. AGs are terminated
viii. Local Message Space spawns Activity Evaluators (AEs) to monitor and record the progress of the activity
ix. AEs update the Subtest swarms through local RA and CMS with updates until the activity is over.
x. Once the Activity is over, AEs via RA and CMS update the General Evaluator with the evaluation results from the subtest swarms.

B. Agents in a Subtest Expert Swarm

All Subtest Agents Swarms function in similar manner. Each of these Swarms has a common combination of agents as shown in Figure 6. The requests agents are the only agents in these swarms that contact with the external world. The other agents inform the request agents their need of external contact via the local message space so that request agent can facilitate them in their task letting these other resource agents to engage in their remaining tasks asynchronously. However, as it was described earlier, the major responsibility of these kinds of swarms is to create, monitor, and evaluate their rightful and respective subtest (e.g.: Block Design, Similarities, Digit Span, etc.). The abstract flow process flow of these swarms is shown in Figure7.

Central Message space spawns the relevant subtest request agents as a result of an OneShotBehaviour. Along with this spawning, the request agent communicates with local message space which then spawns the necessary local agents to carry out the subtest. The Subtest Constructors generate the subtest and prepare it for the user interaction. Likewise the

Subset Monitors keep track of the change of states in the subtest and finally, Subset Evaluators evaluate the subtest with reference to WISC-IV standard evaluation [1, 3].

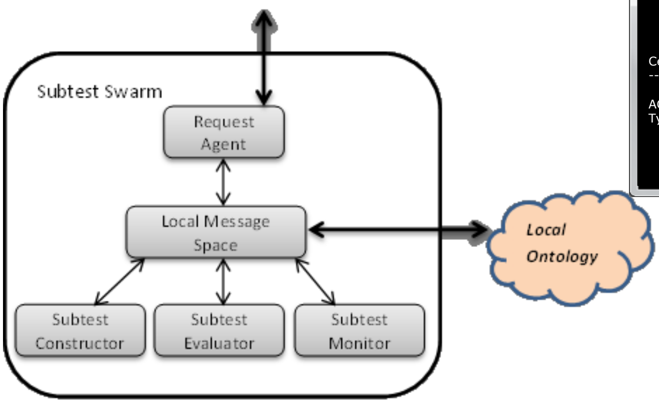

Figure 6: Agents Composition in a Subtest Swarm.

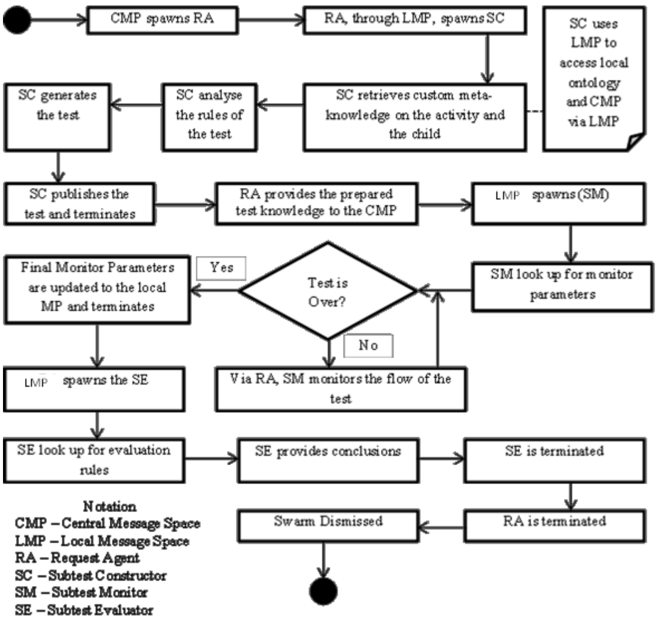

Figure7: Abstract process Flow in a Subtest Swarm During a sub test.

C. Central Message Space, Communication, and Other Special Swarms

Central Message Space is the heart of this agent network through which the swarms exchange messages back and forth. Central Message Space itself has Request Agents to listen to incoming messages from out world – the web service agent, a local message space for maintaining the communication inside this swarm of agents and resource agents for processing work specific for the central system (Course Conductor, Class Admin, and General Evaluator) and for redirecting the messages back and forth via the request agents to destination swarms. The following figure shows the admin information screen of Central Message Space which is updated constantly for troubleshooting and maintenance.

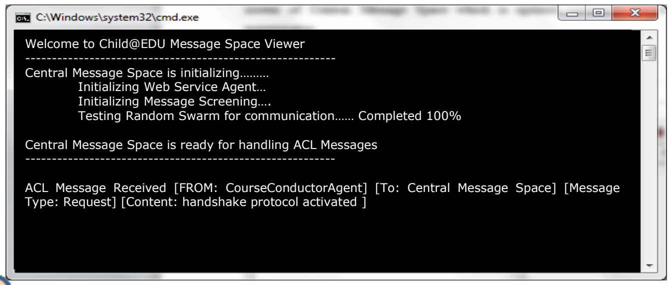

Figure8: central Message Space Message Viewer

The communication between agents happens in the means of ACL messages. ACL messages define a common standard for all the agents to communicate with each other without proposing additional overhead for the developer.

The other two agents: course conductors and general evaluators have been implemented as straight forward as the resource agents of central message space. Especially the general evaluators communicate with Subtest swarms through central message space to retrieve evaluations from each of the carried subtests.

The evaluation rules are derived from WISC-IV specification [1]. Few important rules were stated before and it is recommended that the readers of this paper follow the WISC-IV specification thoroughly as the ontologies model the rules and conditions of WISC-IV as they are stated in that documentation.

D. Screenshots of Graphical User Interfaces

Figure9: GUI screenshots of when a child involves in a writing activity

Figure10: GUI screenshots of when a child involves in a reading activity

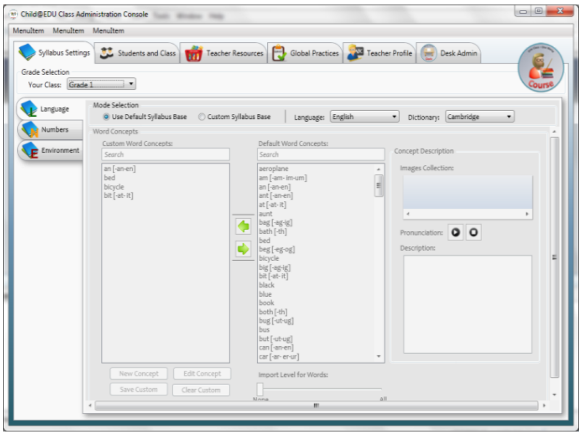

Figure11: Administration Console for Teachers

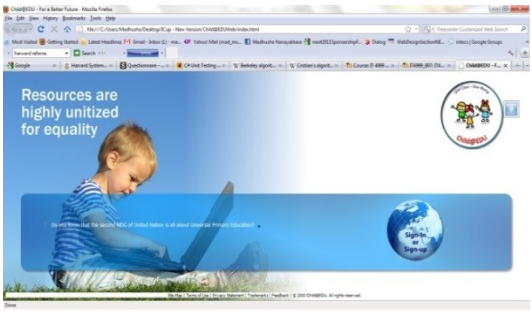

Figure12: Home Page of the web site

7. Evaluation

The functionality and the quality of Expert Agents Module have been tested in the general testing means of CHILD@EDU system prototype. The testing carried out under the evaluation can be divided in to mainly 4 categories. They are as follows.

i. Unit Testing: These tests have been carried out in programming level of the prototype system to ensure a reliable screening of users to determine the final outcome of the users
ii. Integration Testing: Testing application layer and user interfaces separately.
iii. Functional/System Testing: Testing overall system requirements.
iv. Acceptance Testing: User perspective of the system's quality.

Being Specific to Expert Agents Module and for testing Java code, JUnit 4.x has been used.A large number of Unit test cases have been used in every part of the module to ensure higher quality in unit level.

The best way of assuring the quality of Expert Agents Module was to test it as a whole unite through Integration testing. Integration testing has been carried out while fixing different modules together to form up the prototype of Child@EDU. Especially, when integrating different layers different strategies have been used. The most significant of all is the integration of Expert Agents Module with the web service.

Figure13: Web Service Tester Interface

In Summary, The importance of this research and development project is that it has also run a real acceptance test using children of related samples of interests. The quality of the analysis of the inputs and the produces results clearly reflect the quality of the Expert Agents Module. In other worlds, the result of the acceptance test of CHILD@EDU is a good determinant of the quality of its Expert Agents Module. Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV) has been a worldwide standard testing mechanism for children's learning and intelligence abilities.Its specification comes with proven reliability and dependability. The following table shows the default WISC-IV classification of children depending on the general scores of all tests for WISC-IV [1, 3].

Table 1: Traditional Descriptive System for the WISC-IV

Traditional Descriptive System for the WISC-IV	
Standard Score Range	**Description of Performance**
130 and above	Very Superior
120 to 129	Superior
110 to 119	High Average
90 to 109	Average
80 to 89	Low Average
70 to 79	Borderline
69 and below	Extremely Low

As CHILD@EDU has also adopted this evaluation mechanism, the user tests have been carried out involving 25 children. First, the children were insisted to face three consecutive testing cycles of WISC-IV [1] manual testing. Then the same sample of children was driven to face CHILD@EDU primary education platform on which the tests were run cognitively. Table 2 and Table 3summarize the results.

It was clearly visible that during the first cycle of testing in CHILD@EDU platform the children clearly had a negative deviation. This was due to the fact that the new environment of testing made a mental retaliation and a sudden shock in them. However, during the third cycle, the effect was very similar to WISC-IV manual testing [1, 3]. Thus, it can be concluded that CHILD@EDU with proper adaption to the system can function almost as the manual way speeding up the process and effectively utilizing the resources.

Table 2: WISC-IV manual testing

WISC-IV Manual Testing on 25 Children							
	Very Superior	**Superior**	**High Average**	**Average**	**Low Average**	**Borderline**	**Extremely Low**
Cycle 1	1	2	4	8	5	3	2
Cycle 2	1	2	5	7	5	4	1
Cycle 3	1	2	6	7	5	3	1

Table 3: CHILD@EDU Testing

CHILD@EDU Testing on the same 25 Children							
	Very Superior	**Superior**	**High Average**	**Average**	**Low Average**	**Borderline**	**Extremely Low**
Cycle 1	0	2	2	7	7	4	3
Cycle 2	0	1	6	6	5	5	2
Cycle 3	1	3	5	6	6	3	1

In addition to this, a printed questioner was distributed among a sample of teachers numbered 20. In summary, 80% of the teachers agreed that CHILD@EDU can become a very useful tool for primary education. Moreover, the test results clearly have shown that the results obtained through the process that extended the framework is accurate, complete, and robust.

8. Limitations and Further Work

A. Limitations of Expert Agents Module

CHILD@EDU[11] as a digital knowledge system in a very sensitive domain has faced many challenges during the design time as well as in the implementation. Especially, the following limitations can be noticed in its Expert Agents Module.

- ***Strongly Coupled with WISC-IV:***WISC-IV is a widely used standard for modeling primary education activities and evaluating children. Yet, it is neither the only one nor the one used by everyone. CHILD@EDU agents are, more or less, WISC-IV agents. Therefore the approach propagates limitation in flexibility of using other standards.
- ***Ontologically Static Agents:***The knowledge structures of these expert agents are static whereas their knowledge on those static structures increases with the time. This aspect limits the achievement of full dynamicity of agents. If the agents were fully dynamic (in contrast to being partially dynamic) and with fully learning capacity (in contrast to partially learning agents), the capability of the system could have reached further heights.
- ***Static and Strong Dependency on JADE:*** Currently CHILD@EDU Expert agents solely run on JADE. Therefore the limitations that jade faces in terms of memory management, capacity handling, and web service connectivity, may also affect CHILD@EDU expert agents and the swamp expansion. Especially, CHILD@EDU implements its own web service access library in order support the complex tasks of its users, which communicates with JADE in indirect ways.
- ***Agents to Support Teachers, Parents, and other Parties:*** CurrentlyCHILD@EDU does employee agents specifically as assistants for the teachers. Currents agents are just friends of the Children and the workforce of the system. Therefore, as of now, the solution limits theabilities of other users over this knowledge system.
-

B. Further workof Expert Agents Module

CHILD@EDU's capabilities and capacity solely depend on its Expert Agents Module. Further work of this module may reflect back as an increment of the efficiency and effectiveness in the processes of primary education. As of now, it completely focuses on children and providing them an immersive experience in learning while silently evaluating them. The following list is suggested as further work of this module.

- ***Freelancing Agents:***All the agents in CHILD@EDU expert agents' pool is allocated a specific task. As further development of the pool, it is suggested to include additional agents to support dynamic miscellaneous functions.
- ***Intelligent Agents to Support Speech-to-Images Graphical Guidance:*** With the power of flexibility built-in to CHILD@EDU and its Expert Agents Module, it can be further developed as means of add-ons. This suggesting feature is one such add-on that can heavily heighten the learning experience of children. Respecting the famous quote "A picture worth a thousand of word", it is suggested to visualize the speech of teacher and even child when engaged in various activities.
- ***Agents to Support Teachers:***As the closest next step, it is recommended to include agents as assistants to teachers adhering to WISC-IV standards.
- ***Agents with Common Basic Fully Dynamic Ontology:***Having a basic ontology, with programming language specific technologies like Reflection, can be used to create fully dynamic agents: agents that change its own knowledge definition and grow intelligently.

9. Conclusion

CHILD@EDUis a good solution to some critical problems faced in Primary Education. It was said that only a specialized approach can be adapted to solve the following issues found in modern primary education systems.

- One Teacher-One Child education is not being globally possible
- All Teachers or parents are not being experts in child psychologists or physiatrists.
- Effective report generation of a child's educational attributes
- Supporting Effective Student Centric Teaching
- Integrating Parents, Education Institute, and Children
- System of Systems is needed for efficient government decision making

The challenge of addressing all these issues is to find a modern-day technology approach to produce a system which contains entities that may work is met effectively by CHILD@EDU by using the powerful ***multi-agent technology*** [11]. It introduces the agency of expert humans in specific micro elements of an effective learning of a primary child in software environment, In simple words, software agents function as unique expertise on 5 elemental characteristics of effective primary learning with reference to Wechsler Intelligence Scale for Children–Fourth Edition (WISC-IV) Standard [1, 3]. Moreover, it adopts an evaluation mechanism from WISC-IV to effectively monitor and evaluate the learning and intelligence attributes of children.

CHILD@EDU Expert Agents module is where it facilitates the inhabitance of distributed work-delegated, agents each handling an atomic task. These tasks ultimately help the system to generate educational and testing activates for the children, monitor them, or evaluate their progress. It is a very effective way of introducing expert computer agency in the field of Primary Education. The robust design of the expert agent module enables fast communication, increased cardinality of work and users, and accurate decision making.

Expert Agents Module too face many limitation inherited by technology as well as by the design itself like any other solution in the world. Out of these, the main attention should be given to develop dynamic learning agents so that the system itself can evolve and expand itself horizontally without external interference. Further, freelancing agents in addition to the agents that are strongly coupled with WISC-IV could increase the systems performance drastically.

CHILD@EDU could overcome most of the problems that could be found in the compared systems stated in Section II. Especially, with expert psychological focus on each child and by offering personalized/ customized course in general classroom, it demonstrates an immersive capability of working with sensitive aspects of children than the mentioned systems while effectively evaluating and producing reports. Unlike most of those systems, CHILD@EDU supports the primary education system for seeing the true attributes of children and enforces better and effective decision making.

Acknowledgment

We show our sincere gratitude to Professor A. S. Karunananda, Professor, attached to the Department of Computational Mathematics of the Faculty of Information Technology,University of Moratuwa, Sri Lanka and Dr. N Kumaranayaka(Psychologist), CEO of HOPE foundation for the extensive guidance

provided and the effort made to increase our intellectual level on this particular subject of research.

References

[1] Aurelio Prifitera, Donald H. Saklofske, Lawrence G. Weiss, 2008, *WISC-IV Clinical Assessment and Intervention*: Academic Press is an imprint of Elsevier

[2] CSM Report, 2002. *Education personnel Information System for Orissa Primary Education Programme Authority,* India.

[3] Dawn P. Flanagan, Alan S. Kaufman, *Essentials of WISC®-IV Assessment*: John Wiley & Sons, Inc.

[4] Dongdai Zhou, 2010, *SOA-based Education Information System Interoperability Model*. Journal of Information & Computational Science

[5] Education .com, *The 8 Characteristics of Motivated Kids*. [Online] Available at:

http://www.education.com/magazine/article/go-for-it-motivation/

[6] Gomez-Perez A., *Ontologies: Theory, methods and tools*. [Online] JoergDiederich (L3S Research Center). Available at: http://rease.semanticweb.org/ubp/PUSH/search@srchDetailsLR;jsessionid=4DA06220357A7DE31C2F139DBD8BF53C?lrID=lr-lear-diederich-1171378902347

[7] Great Education, *The Importance Of Elementary Education*. [Online] Available at: http://www.gr8-education.com/the-importance-of-elementary-education.html

[8] InfoDev, October 2006, *Education Management Information System: A Short Case Study of Mozambique*. WORKING PAPER

[9] *Java Agent DEvelopment Framework*. [Online] Available at: http://jade.tilab.com/

[10] John Eagles, 2001*, Making Classroom Talk*. Global Education Center

[11] M.S. Nanayakkara, K.A. Dilini T. Kulawansa, 2011, Agent based Primary Education Supporting Platform: ITRU Research Symposium - 2011

[12] Magenta Technology, *Multi-Agent Technology*. [Online] Available at: http://www.magenta-technology.com/web/guest/multi-agent-technology

[13] Miles Berry, 12th October 2005, *A virtual learning environment in primary education*

[14] Wikipedia (April 2010), *ASP.NET*. [Online] Available at: http://en.wikipedia.org/wiki/ASP.NET

[15] Wikipedia (April 2010), *Java Platform, Micro Edition*. [Online] Available at: http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition

A Multi-agent based Approach to simulate Uncertainty of a Crowd in Panic with Sharable Ontologies

Wimal Perera[1], Asoka Karunananda[2], Prasad Wimalaratne[3]

[1,2]Faculty of Information Technology, University of Moratuwa
Katubadda, Sri Lanka
[3]University of Colombo School of Computing
Colombo, Sri Lanka
[1]wipelk41@gmail.com, [2]asoka@itfac.mrt.ac.lk, [3]spw@ucsc.cmb.ac.lk

***Abstract*— Crowd simulation is listed under many practical applications in computer industry; such as safety modeling, pre-planning building architectures, urban modeling and entertainment software. Most of these existing simulations are created by implementing computer algorithms based on extending deterministic models such as particle systems, clustering, cellular automata and fluid motion. However, extending a crowd model to simulate uncertainty of crowd behaviour during panic still remains a key challenge; since a computer algorithm approaches a solution by parameterizing predictable elements within a problem.**

It is evident from literature about the proven success of multi-agent technology behind modeling complex systems; comprising of many distributed entities interacting with each other and operated under lot of uncertainty. Thus it can be postulated that multi-agent technology provides a basis to model the uncertainty raised within a crowd during panic. Our proposed solution simulates this uncertainty by considering evacuation of a crowd from a building during fire. Each individual in the crowd is modeled as an agent associated with a local ontology. The local ontology of an agent is a collection of rules, representing the knowledge known to each individual prior to occurring fire. Rules embedded within local ontologies are shared among individuals as they interact with each other. As a result non-anticipated global behaviours arise within the crowd leading to emerging uncertainty during fire. Output of the system is a visualization of crowd behaviour during fire with recorded statistics. The statistics recorded during each simulation session indicate evacuation related information per each individual; providing a basis for evaluation by comparison with real world observations.

***Keywords*— Emergence, Multi Agent Systems, Sharable Ontology, Crowd Simulation, Crowd Behaviour**

1. Introduction

A crowd can be defined as a large group of individuals residing within some physical environment[1]. Crowd modeling is one of the most interesting research areas in computer simulations; focusing on how to simulate the behavior of each individual, in order to generate a resultant behaviour for the entire crowd. Owing to their different approaches, existing studies on real world crowd behaviour under different contexts[4][6][7] and developing a crowd simulation model based on recorded observations from these studies[1][2][7] are quite distinctive to each other. Yet a difficulty lies behind extending an existing crowd simulation model to support panic; where every individual must quickly adapt with unpredictable surroundings [1][2]. The major reason for this challenge is that; conventional approaches behind developing existing crowd simulation models concentrate on forming a predefined method or an algorithm by observing a predictable pattern in the crowd for a given scenario.

Panic behaviour of a crowd during an emergency is a decentralized process involving lot of uncertainty which can't be predicted beforehand. An example is how people in a building would immediately evacuate during a fire [2]. In most cases a single crowd may cluster into asymmetrically distributed subgroups. This happens due to arbitrary actions done by individuals for survival during fire in which they even aren't aware of. Therefore the real problem arises when attempting to model uncertainty of a crowd in panic by using a deterministic approach such as algorithms.

Multi-agent technology [9] provides a basis for developing solutions for decentralized systems involving a lot of uncertainty. Special characteristics of multi-agents such as communication via message passing combined with emergence could be used as a base to model an unpredictably changing physical environment with uncertain individuals. Therefore, individuals in a crowd can be modeled as agents having certain perception about their vicinity with their own limited knowledge? These individuals can be made to communicate among themselves, share each one's incomplete knowledge and let overall crowd behavior "emerge" due to interaction [9][10].

The rest of this paper is organized as follows. Section 2 reports observations from real world crowd behaviour studies during panic and reviews existing crowd simulation models. Section 3 states relevance of multi-agent technology with crowd simulation. Section 4 describes multi-agent based knowledge sharing approach behind emerging uncertainty in a crowd during panic. Section 5 elaborates design of the multi-agent based crowd simulation model. Section 6 describes about the implementation. Section 7 focuses on evaluation while Section 8 stating final conclusion with further work.

2. Crowd Behaviour, The State of The Art

This Section attempts to, (i) discuss general observations from studies based on real world crowd behavior under panic and (ii) study underlying concepts behind developing a crowd simulation model; hence identifying the limitations behind extending a crowd simulation model to support uncertainty of crowd behavior during a panic.

A. Studying Crowds under Panic: General Observations

Studies by Helbing et. al. [3][4] state that following characteristics can be found within a pedestrian crowd under panic. (1) People try to move considerably faster than normal, (2) Individuals start pushing, and interactions become physical. (3) Moving (especially passing a bottleneck) becomes uncoordinated. (4) Arching and clogging are observed at exits (5) Jams build up (6) Physical interactions in jammed crowds build dangerous pressures able to bend steel barriers or push down brick walls (7) Escape is slowed by fallen or injured people acting as 'obstacles' (8) People show a tendency towards mass behaviour (doing what other's do). (9) Alternative exits are often overlooked or not used in escape situations.

Helbing et. al. [4] further reveals that, the strategy for escape from a smoke-filled room, involves a mixture of (1) individual behaviour and (2) "herding" instinct. Individual behaviour refers to irrational behaviour within individual in panic, especially with scarcity of resources (limited time and limited space for exit). Collective "herding" instinct refers to mass behaviour (i.e. tendency to follow what others do, without thinking is it correct or not). Together, these 2 tendencies cause an un-coordinated motion among individuals during exit, which is the root cause for developing jamming, arching, clogging plus injuries due to crowd interactions.

A study by Nishinari et. al. [7] reveals that the behaviour within group of a ants, reaching a certain destination exhibit similar characteristics to the individual behaviour of humans. Just like each ant used to follow its predecessor ants; humans too tend to follow other humans in a crowd for efficient and safe walking. Ants communicate with each other by dropping a chemical called "pheromone" on the ground as they move forward. The pheromone sticks to the ground long enough (i.e. before getting evaporated) for the other following ants to pick up its smell and follow the trail [8]. In terms of an emergency; humans also tend to follow trails left by predecessor humans for reaching emergency exits or safe places. Similar to pheromone emitted by an ant getting evaporated after some time, trace opened with movement of a predecessor human is closed in quick time due to the rapid movement of individuals and entities in the environment during panic.

Fleya et. al [6] argues that for an individual in a crowd under panic making a decision between adherence to social norms versus going by selfishness; is not only based on scarcity of resources but also pre-known historical context. The study was based on recorded survival statistics during sinking of 2 ships, Titanic and Lucitania. For evacuation from Titanic, social norms took over selfishness. Since it was pre-stated as "Titanic is unsinkable", passengers anticipated that they will be rescued. During entering life boats; first class passengers, women and children were preferred over others. Abidingby social norms made Titanic sank slowly(on 14th April 1912 for 2hrs and 40mins). Lucitania sank after Titanic (on 7th May 1915 in 18 mins). For Lucitania, scarcity of time along with pre-known historical context on lesser chance for survival, made no emerging in social norms, but selfishness among passengers.

B. Crowd Simulation Models

With the overview from Section 2 Aon real-world crowd behavior during panic; this section focus on existing crowd simulation models and their underlying concepts.

C. Reynolds [11] developed a method for simulating a crowd such as school of fish/flock of birds using particle systems, where each individual in the crowd is treated similar to a single particle within a particle cloud. The major focus of this simulation model is about the motion patterns of the crowd by leading it towards reaching for a specific location as the final target, similar to particles in a cloud flowing along the wind direction. Therefore the motion of a specific individual is always directed towards achieving the final destination.

D. Terzopoulos [12] developed a method to animate crowd behaviour based on modelling each individual in a crowd with capabilities to percept its surrounding environment. The behaviour of each individual is achieved by defining a predefined set of behaviour patterns for a predefined set of sensory inputs. For instance, modelling a school of fish in water; sensory inputs are temperature and vision (to identify surrounding objects). Predefined behaviour patterns are finding food when hungry, avoiding areas with high temperature and avoiding collisions with other fish.

S. R. Musse and D. Thalmann [1] proposed a method to simulate human crowd behavior by identifying inter-group relationships between individuals and mapping these relationships within local and global set of goals formed according to the needs of each individual. This research uses people who visit a museum as its model example. In a museum artifacts belong to certain groups are categorized and separated into areas/rooms. People with same interests gather around these separate rooms/areas forming clusters within the crowd, where each cluster of people has similar interests.

Cherif Foudil [2]argues that the behavior of any crowd is significantly different in terms of a normal situation comparing to a situation in panic. This argument is supported by Helbing's study [3] in terms of simulating pedestrian crowds, where it is clearly identified that crowds demonstrate completely different characteristics in normal and panic situations.Therefore according to Foudil's model, in normal situations a crowd consists of groups of individuals and crowd behavior is distributed to groups followed by individuals. However in terms of panic situations, the concept of a group is lost and all individuals are expected to work for themselves based on their basic instincts and group around exit points.

Further Foudil [2] argues that modeling a crowd as a collection of particles (where each individual is a particle) improves realism in panic situations comparing to fluid-based approach (where crowd movement is considered analog with the fluid flow) followed by some crowd simulation models [3]. The reason is that suppose there are 2 exits in a room, clogging individuals among one of the exits occur due to their unexpected behavior during panic, whereas in a fluid-flow model will divide the loads of the crowd uniformly among the 2 exits; so that this won't reflect the realistic behavior.

Nishinari et. al. [7] extends their study of crowd behavior (described under Section 2 A) and proposes cellular automata based grid model with the concept of "pheromones". The grid represents spread of pheromones within an area. Each cell in the grid either has a pheromone or free of pheromones. Further each cell will have one or more ants. Any ant decides its next direction of movement based on spread of pheromones in its neighboring cells. The argument is that just like temporary traces are created when pheromones fall and disappears when pheromones evaporate, in a similar manner temporary routes towards exits are created and vanished within an evacuating crowd in panic.

C. Multi-Agent based CrowdSimulation Models

With the dawn of multi-agent technology [9] (discussed under Section 3) as a new paradigm to model intelligence in simulations, several attempts have been made to simulate crowd behaviour by modelling individuals within a crowd as agents.

MASSIVE [16] is a commercial software product used for generating crowd related visual effects and autonomous character animation. It has been originally built for a crowd simulation scenario related with "Lord of the Rings" film series and since adopted as well-known crowd simulation software in film industry [16]. From the available information [16] about the internal architecture of MASSIVE, each individual in a crowd is modeled as an agent, and variety of configuration options are available to configure rules for the agents based on conditions related with his/her vicinity. Further within a 3-dimensional virtual space, the user can configure the paths and directions which a certain group of people must follow. Hence the software will simulate the group behavior of individuals based on the motion paths being pre-configured. Therefore the agent implementation of MASSIVE can be concluded as a goal-based agent [17] where the goal is to follow to the destination via the closest defined path.

Ana Luisa et. al. [18] proposes an agent-based model to simulate a pedestrian crowd in a corridor as a tool to observethe behavior of human crowds in, routine and crisis situations. This simulation model combines the ideas from kinetic theory of living systems with ideas from the field of computational agents. Kinetic theory is used to setup global parameters for the crowd simulator such as crowd density, step size and defined paths. In Luisa et. al. [18] model; based on parameters such as pedestrian density and time left to reach the destination, the goal-based intelligent agent [17] (i.e. representing an individual in the crowd) will choose the most appropriate path among several available paths. Based on the perception about the vicinity, the individual (i.e. goal-based agent) might anticipate or give the chance to an action done by another individual, prior to executing his/her selected action.

D. Key Challenge in Extending Crowd Simulations to support Panic Behaviour

For simulating crowd behaviour, different models are proposed based on different concepts with the intension of reproducing the same observations received by studies related to real-world crowd behaviour. Such discussed existing crowd simulation models included; (i) particle systems based crowd models [5][11], (ii) fluid flow based crowd models [2], (iii) cellular automata based crowd models [7] and (iv) clustering a crowd into groups based on similar interests [1]. These models look at a crowd from a macroscopic view (i.e. an overview of the crowd as a whole) and have developed algorithms to simulate crowd behaviours based on the respective approach. Although the above crowd models can simulate observations of crowd behaviour related with normal situations, their capabilities are limited in terms of simulating uncertainty of crowd behaviour during panic. The rationale behind this limitation is that; a computer algorithm always approaches a solution by parameterizing predictability within a problem, whereas crowd behaviour in panic involves a lot of uncertainty.

Existing Multi-agent based crowd models focus on modelling crowd behaviour from individual's perspective (i.e. comparing to macroscopic view/overview of crowd followed by conventional crowd simulation models). The approach is to build intelligence into the agent (i.e. individuals) by means of a pre-defined rule set and make the individual (i.e. agent) more perceptive to the changes of his/her surrounding environment. The individual selects the most appropriate action based on the current status of surroundings and his/her ultimate goal to be reached (i.e. goal-based agent [17]). The goal is predefined and given to the individual by means of proposed path to be travelled towards the destination. Thus goal-based agent approach proposed by existing agent-based crowd models doesn't fit well into the context of simulating uncertainty of a crowd in panic; since during panic, there aren't any predefined goals except survival.

Therefore the key challenge is how is it possible to simulate uncertainty of crowd behaviour during panic?

3. Application of Multi-agent Technology for Crowd Simulation in Panic

Multi-Agent Systems (MAS) [9] proposes a new approach to develop intelligent software comparing with conventional AI approaches (such as neural networks, expert systems and fuzzy logic). A MAS is made up of a large number of agents which interact with each other. Each agent has incomplete information and insufficient capabilities for solving a problem. In MAS there is no system global control (like in neural networks, expert systems) and data is decentralized with asynchronous computation. The key feature of MAS is that although each agent (or most of the agents) possesses incomplete information and insufficient capabilities; it's the interaction between these agents which enables a problem to be solved by each one of them. Due to interaction between agents within MAS, each individual agent receives ability to reason about non-local effects based on its or other agents' local actions, thus forming expectations based on the behaviour of others [9]. This results in emergence, where "emergent intelligence" is defined as ability to arise effective solutions to problems under conditions of uncertainty [10].

For applicability of above MAS concepts to simulate crowd behaviour in panic; consider a sample scenario about individuals evacuating from a building during a fire. There can be different personalities among individuals in the context of escaping from a fire; (1) normal person, (2) fire-warden and (3) security personnel. A normal person is not aware or minimally aware in terms of the quickest steps to be taken in case of a fire. Therefore he/she would have not more than common sense. However a fire-warden appointed for each floor would have a specific training on actions to take during a fire such as operating relevant fire extinguishers. On the other hand security personnel would not have a specific training for fire, but they will have access to information about all individuals in the building in each floor in a given moment, which a fire-warden is not aware of.

All these 3 personalities operate based on incomplete information. In terms of multi-agent technology we can model each individual as an individual agent and each individual will have an "initial local ontology" (i.e. an initial knowledge base) according to his/ her personality (i.e. a normal person, a fire-warden or a security personnel); regarding how to operate during a fire situation. But as time passes these agents will learn from each other (i.e. they will share each other's local ontologies/knowledge) and ultimately manage to solve the problem of evacuating from the building during fire, by emerging the crowd behaviour anticipated during a real fire situation.

4. An Approach with Multi-Agent based Knowledge Sharing for Crowd Simulation

The intention of this section is to discuss our approach behind forming a solution to simulate uncertainty of crowd behaviour during panic by enforcing knowledge sharing with multi-agent technology. The approach is initiated by defining input, process and output for developing our multi-agent based crowd simulator followed by identifying the set of features that can be provided along with the range of end-users who will be benefited from these features.

The input for our multi-agent based crowd simulator is the Environment Setup Information, which consists of 3 major input types (i) Fire Related Information, (ii) Exit Point Related Information and (iii) Individual Related Information. Fire Related Information is about the set of fire sources including origin point (i.e. location in the building) of each fire source corresponding to initially starting fire. Exit Point Related Information is about the list of exit locations (either regular doors or emergency exits) within the building.

Individual Related Information is the list of individuals currently residing inside the building. Each individual will consist of (i) his/her location in the building, (ii) individual type (whether he/she is a normal person or a fire warden), (iii) perception bounds value indicating the maximum range which he/she can directly communicate with other individuals and (iv) an assigned initial confidence level.

Upon request of the desired Environment Setup specified by the input, relevant agents will be created for (i) simulating individuals in the crowd and (ii) simulating spread of fire.

Output of the system include (i) rendering a visualization based on changes happening to individuals in the crowd and (ii) recording statistics based on time of exit for each individual from a given exit point of the building.

The main features provided by our crowd simulator include; (i) Configurable Environment, (ii) Statistics collected periodically in terms of individual exit and (iii) Graphical view of Crowd Simulation over time. A configurable environment includes defining; (a) multiple fire sources with their initiating locations within a building, (b) individuals with different personalities (i.e. normal person, fire warden, security guards etc.) within different locations of the building and (c) locations of exit points in the building. The simulator logs the exit of each individual from a given exit point against the time elapsed. This log of individual exits can be later used to (a) compare the validity of the simulation with available data for a real scenario, (b) derive inferences about the consequences in case a real scenario occurs based on the configured environmental setup. Finally the graphical view includes visualization on how individuals would exhibit panic behaviour and ultimately make their exit from the building as the fire spreads.

Our crowd simulator can be used by Building Architects and Security Managers. For instance an architect can be benefited by using the simulator to infer the behaviour of crowds based on the number and position of emergency exits in the system and decide the optimal places to establish the exit points beforehand during building design. A security manager can take decisions such as the numbers and placements of fire wardens, security cameras and security guards for optimal preparation of an existing building infrastructure in a panic situation.

5. Design of Multi-Agent Based Knowledge Sharing Crowd Model

This Section elaborates our approach further as a comprehensive design that can be used as the foundation for implementing a simulation model demonstrating a crowd in panic. Figure 1 provides an overview of the Multi Agent-based design for our proposed Crowd Simulation Model.

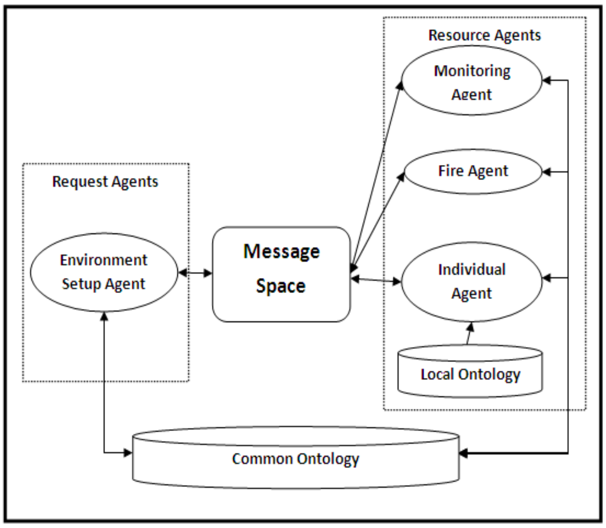

Figure1: Multi-Agent based Design for Proposed Crowd Simulation Model

Note that this design is constructed based on the resource-request-message-ontology architecture [17] proposed for developing multi-agent systems; indicating request agent types, resource agent types, ontologies and message space.

A. *Instantiation of Agents*

In any scenario there will be 1 instance from Environment Setup Agent type and 1 instance from Monitoring Agent type.

The Environment Setup Agent takes the input to the system (i.e. "Environment Setup Information" as described in Section 4) and triggers creation of relevant agent instances for all other agent types based on input.

The Monitoring Agent receives messages from individuals and fire sources within the simulation and records statistics for these messages based on the timestamp.

The number of instances created from Fire Agent type will be equal to the number of fire sources specified under "Environment Setup Information" provided by the input (as described in Section 4). The intention behind a Fire Agent is to simulate spreading of fire and increase in fire severity as time progresses.

Similarly the number of instances created from Individual Agent type will be equal to the number of individuals in the crowd as specified under "Environment Setup Information".

B. *Common Ontology*

The Common Ontology contains 3 types of information; about fire sources, individuals and exit points in the building. In other words, it is a representation of "Environment Setup Information" provided by the input. Note that, this Common Ontology is initialized by the Environment Setup Agent. Later Fire Agents and Individual Agents will update the Common Ontology periodically. Therefore we can conclude about the Common Ontology as a snapshot of the crowd simulator (i.e. individuals and fires) at a given time.

C. *Message Space*

"Message Space" provides the infrastructure for our Multi-Agent based Crowd Simulator to enable asynchronous messages between agents; which is an essential feature of a Multi-Agent System. Any message sent by any agent is dispatched to message space. The Message Space stores any message (sent by the sender agent) until its recipient agent queries and retrieves the message completely.

D. *Individual Agent*

During execution, any Individual Agent will refer to his/her own "Local Ontology" in addition to the Common Ontology. Further an individual agent may also update his/her Local Ontology, which implies learning over time from other individual agents. This local ontology of each individual is "initially" created according to the personality type for each individual. For example in case of our example scenario of evacuation from fire, we may think of 2 different types of personalities; normal person and fire warden. Therefore there are 2 templates to represent the pre-known knowledge of a normal person and a fire warden prior to occurring fire. A template is a collection of rules representing general actions those are possible by an individual having the personality focused by the template.logy maintains knowledge related to agents in the system.

E. *Local Ontology of an Individual Agent*

Each Rule within a local ontology of an agent consists of the components illustrated as in Figure 2.

ID is used as a unique identifier to represent each rule.

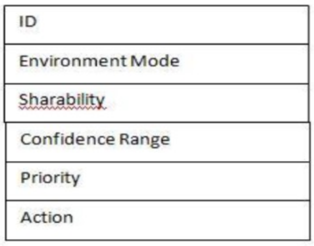

Figure2: Contents inside a Rule

Environment Mode indicates the specific type of environment, which the individual must consider about applying these rules. The environment mode of the system will vary with time based on the changes of the environment encountered due to various activities happening in the system. For instance, in the case of evacuation from fire during a building, there can be 2 environment modes; Stable Mode (before fire starts) and Fire Mode (after fire has started and during fire).

Sharability indicates about the extent which the action specified by a rule can be shared by other personalities. For instance an action such as running towards an exit point can be done by any person. However an action such as using fire extinguisher can be done only by a person who has undergone some form of fire avoidance program (such as a fire warden).

The individual confidence level of each person will vary with time as the fire progresses. For instance the confidence level of a person is very low, when the fire is severe and the person is very close to fire. On the other hand the confidence level of an individual is high when fire is not severe and the individual is far away from fire. The Confidence Range of a rule indicates the minimum and maximum levels of confidence, which a particular individual will attempt selecting a given rule for execution.

The Action indicates how an individual will behave if he/she selects a rule based on rule selection criteria.

1) Rule Selection Criteria: Each individual agent will periodically select a rule (containing an action), from his/her local ontology. This rule selection will be based on; (i) Environment Mode, (ii) Confidence Range and (iii) Priority. The agent will select a rule which matches the current Environment Mode along with his/her current confidence level. If there is more than 1 rule matching these criteria, the agent will prefer the rule with highest priority.

2) Neighbour's Rule Acceptance Criteria: After executing the action of a particular rule from the ontology by a particular individual, the other individuals in the vicinity will come to know the action taken by this individual. In other words the latest action taken by a particular individual is multi-casted to all individuals in his/her vicinity. Others will try to incorporate the rule, just executed by this individual, if they are interested about the action came out of this rule. This interest of an individual towards a neighbour's action depends on; (i) Neighbour's Confidence Level, (ii) Environment Mode and (iii) Sharability. Obviously for a person to consider about a neighbour's choice, the neighbour must exhibit higher level of confidence, other than the person's own. The Environment Mode assists in finding whether the neighbour is doing something relevant with the current state of the environment. Sharabality indicates whether, what is done by the neighbour; can be done by the individual as well. If all these conditions are met an individual will update his/her own local ontology by adding the rule applied by the neighbour.

3) Emergence with an Evolving Local Ontology: As time passes the exchange of messages between individual agents lead to sharing the "Local Ontology" of each agent among each other, since every agent gets to know the actions done by his/her neighbour. For instance a normal person could learn on what a fire warden does and will update his/her Local Ontology based on what just the fire warden did. This leads to evolving the "Local Ontology" of each individual, leading to learning within each individual. Therefore in a given instance of time, any individual agent exhibits a "better evolved local ontology" adapted towards the panic situation; comparing to his/her "initial local ontology". This evolution of the Local Ontology within each individual agent results emergence of previously not anticipated behaviour patterns when the overall system is observed.

6. Implementation of Multi-Agent based Crowd Model with Sharable Ontologies

This section describes implementation of request/resource agents, message space and ontologies identified during the design; into a prototype that can be used for demonstrating a fire evacuation session for a given input scenario. The overall implementation of the prototype is carried out using JADE (Java Agent DEvelopment) framework [13], which is an open-source platform designed for developing multi-agent applications involving peer-to-peer communication between agents.

A. Communication between Agents in Message Space

Communication between agents in "Message Space" is implemented such that any message being sent by a sender agent is stored in the message space provided by the JADE platform until the relevant receiver agent/agents retrieve the same message. As stated by Table 1; 3 custom message types are used to exchange messages between agents.

Table 1: Message Types Exchanged between Agents

Message Type	Purpose	Message Contents	Sender Agent Type	Receiver Agent Type
FIRE MESSAGE	Simulates sensing a fire by all nearby individuals	(i) Location of fire center, (ii) Current severity of the fire source	Fire Agent (Representing a single fire source)	Individual Agents within the affected vicinity of fire source
INDIVIDUAL ACTION MESSAGE	Simulates an individual getting sense of what his/her neighbor does at a given moment	(i)ID of the current rule being executed, (ii)Current confidence level of the Individual who does the action, (iii)Any useful information for executing the action related to this rule	Individual Agent (i.e. representing a single person) who is doing the selected action from his/her local ontology	All Individuals in the vicinity of the Individual Agent who is doing the action
INDIVIDUAL EXIT MESSAGE	To notify monitoring agent about the exit of an individual from fire with the given timestamp	(i)ID of the Individual Agent, (ii)Timestamp during Exit, (iii)ID of the Exit Location	Individual Agent who just managed to reach the Exit	Monitoring Agent

Table 2: Initial "Local Ontology" of a Fire Warden

Rule No	Environment Mode	Confidence Range	Action	Sharable With Whom (Other Personality Types)	Useful Additional Information to Execute Action	Priority
1	Stable Mode	0% - 100%	Common Sense	With Anyone	-	1
2	Fire Mode	10% - 30%	Run to the nearest safe Exit Point	Normal Person, Security Guard	Locations of all exit points	2

B. Sharing Local Ontologies of Individual Agents

Table 2 describes the example rule set used in the initial “Local Ontology” of a fire warden according to the rule structure discussed in Section 5. Table 3 shows initial “Local Ontology” of a normal person.

Table 3. Initial “Local Ontology” of a Normal Person

Rule No	Environment Mode	Confidence Range	Action	Sharable With Whom (Other Personality Types)	Useful Additional Information to Execute Action	Priority
1	Stable Mode	0% - 100%	Common Sense	With Anyone	-	1

The sample “Local Ontology” configurations (in Table 2 and Table 3) are used in the prototype to demonstrate the concept of sharing ontologies between individual agents, which leads to emerging non-anticipated complex behavior patterns during a simulation session.

A normal person can get to know about the exit points in the building during a fire via a fire warden who exists within his/her neighbourhood. This is made possible since the knowledge (i.e. the 2nd rule in fire warden's initial “Local ontology” about exit points) of fire warden is shared among all individuals in his/her vicinity. Thus this will lead to updating the initial “Local Ontology” of the normal person with fire exit related information. Therefore a normal person will learn from a fire warden in its vicinity leading to emerging a previously non-anticipated behavior in the crowd.

The decision taken by a normal person on whether to update his/her “Local Ontology” will be based on “Neighbour's Rule Acceptance Criteria” (see Section 5). Upon updating his/her “Local Ontology”, an individual agent will pick up the most appropriate rule to execute his/her next action, by selecting this most appropriate rule based on “Rule Selection Criteria” (see Section V) from his/her updated “Local Ontology”.

C. Implementation of Individual Agent

Each Individual Agent has his/her own “Local Ontology”, initialized during the creation of the Agent. In addition to the initialized “Local Ontology”, each Individual Agent will have an initial confidence level, as specified by the input fire scenario followed by a perception bounds value and the Environment Mode.

In terms of assigning confidence levels; the initial confidence level for an individual is assigned based on his/her personality type. In general, fire wardens are more confident for facing an emergency fire situation. Therefore a fire warden will exhibit a higher initial confidence level comparing to a normal person. As the fire progresses the confidence level of each individual agent will be updated based on the severity and location of his/her surrounding fire sources. If the severity of surrounding fire sources is high and fire sources are located nearby; then the confidence level of any individual agent will decrease. In contrast, if the severity of surrounding fire sources is low and fire sources are located far, then the confidence level of any individual agent will increase.

The “Perception Bounds” value indicates the range of other individuals visible to a particular individual. Note that during implementing our prototype we have considered a circle with individual agent's current location being the center and perception bounds value being the radius, as the perceiving area for a person.

The “Environment Mode” indicates how an individual has understood the current status of his/her surrounding physical environment. An individual will behave in different ways when the environment is stable and environment is under fire. Initially the environment mode for an individual agent is set as “Stable Mode”, since a fire has not been started. But as an individual agent receives a fire message (in other words when a person senses a fire in his/her surrounding environment) from a surrounding fire source, the environment mode of the individual agent is set to “Fire Mode”.

Finally the pseudocode on next page describes how the behaviour of an Individual Agent is implemented according to our “Multi-Agent based Knowledge Sharing Approach” discussed under Section 4.

```
WHILE "Confidence Level > 0 (i.e. I'm alive)"
DO
    Search for pending incoming messages
    IF there are any pending messages THEN
        message =: getNextPendingMessage()
        IF (message is a FIRE_MESSAGE) THEN
            Set "Environment Mode" as FIRE_MODE
            Re-calculate my "Confidence Level" based on severity and location of fire source
        ELSE IF (message is a INDIVIDUAL_ACTION_MESSAGE) THEN
            IF confidence level of the other person is higher THEN
                Update my "Local Ontology" based on "Neighbour's Rule Acceptance Criteria" (Refer Section V)
            END IF
        END IF
    END IF

Get the most appropriate Rule from "Local Ontology" based on "Rule Selection Criteria" (Refer Section V)
Execute the "Action" related to the most appropriate Rule
Find all individual agents in my vicinity (based on my "Perception Bounds")
    FOR EACH individual agent IN my vicinity
        Send the Rule associated with my current "Action" as an INDIVIDUAL_ACTION_MESSAGE (along with any useful information helpful for executing the action)
    END FOR EACH

    IF there's a change in my Location after the latest action THEN
        Update my location in the "Common Ontology"
```

IF I've reached any of the exit points THEN
Send INDIVIDUAL_EXIT_MESSAGE for the Monitoring Agent
Terminate myself
END IF
END IF
END WHILE

7. Evaluation

In this Section we present the evaluation of the proposed multi-agent based crowd simulation model. Note that, the prototype developed (as described in Section 6)has been used as the major tool for evaluation. Evaluation was carried out by executing a sample set of pre-designed input scenarios on this prototype and comparing the observed results from simulation sessions with general observations found in similar real world crowd evacuation scenarios.

Statistics recorded from different types of real world fire escape scenarios [14] reveal that, the following general observations are found within a crowd running away from fire; (i) asymmetrical distribution of individuals among exit points, (ii) arching/clogging near exit points and (iii) difference in statistics in case the same scenario is repeated with same environment setup (due to uncertainty in crowd behaviour during panic).

During evaluation as the first step, a sample fire evacuation scenario involving a small number of individuals was modeled using the prototype. The scenario represented a floor in a building with 2 exit points (i.e. emergency exits). Fire was represented as a single fire source, where the fire center is located near to the middle of the floor. There were 10 individuals at the time of starting the fire. Out of these 10 individuals 2 were fire wardens whom aware of the emergency exit points to be used during a fire and the other 8 were ordinary people. This scenario was executed twice (i.e. 2 simulation sessions) by using the prototype. The recorded statistics (i.e. recorded separately for both simulation sessions) were compared with the general observations from real world scenarios.

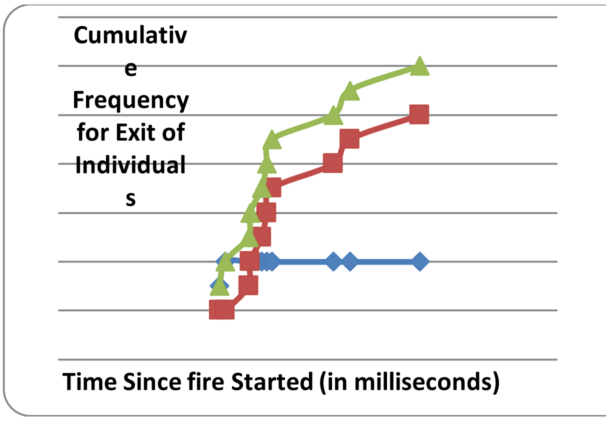

Figure3: Variation of cumulative individual count with time. Scenario 1 (10 individuals), simulation session 1.

Figure 3 is a graphical representation of results from the 1st simulation session with 10 individuals. It is evident that out of the 10 individuals 8 individuals have managed to exit from the 2nd Exit Point. Thus the distribution of the individuals among the exit points has been asymmetrical. Further after elapsing around 60 seconds since the fire has started (i.e. around 60000 milliseconds) several individuals have tried to escape from Exit 2 at the same time. This is an indication of arching/clogging around an exit point during a panic situation, since individuals reach exit points in groups at the same time due to circumstances ruled by uncertainty occurred during panic.

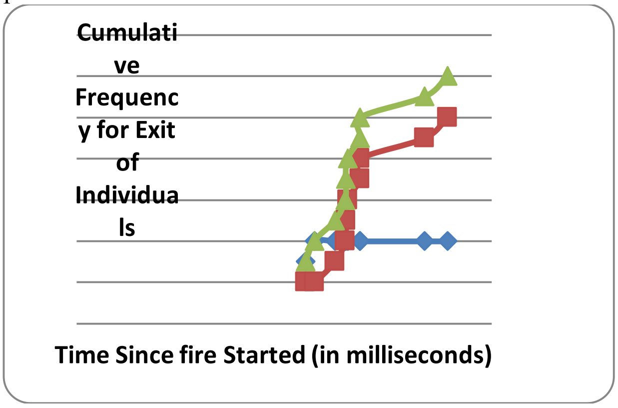

Figure4: Variation of cumulative individual count with time. Scenario 1 (10 individuals), simulation session 2.

Figure 4 shows results for the 2nd simulation session with 10 individuals and also indicates similar patterns like in Fig. 3, such as asymmetrical distribution of individuals among exit points, arching/clogging. However in terms of the arching/clogging effect it is evident that during 2nd simulation session (i.e. Fig. 4) 6 individuals have been subjected to clogging near 60 seconds (i.e. around 60000 milliseconds), whereas during 1st simulation session (i.e. Fig. 3) it was only 4 individuals who were subjected to clogging near 60 seconds mark. This means that although it is the same experimental setup which is executed twice, the statistics have varied slightly between the 2 executions. This is an indication of uncertainty of crowd behaviour during panic via our simulator. Further when comparing Figure 3 and Figure 4 together, it is evident that in both cases the cumulative individual count at Exit 2 is not a gradual increase with time, showing that Exit Points are not fully utilized among individuals which happens due to uncoordinated behaviour occurred during panic.

The experimental setup for Scenario 2 was also similar comparing with Scenario 1 (i.e. single floor of a building with 2 exit points and single fire source), except that the number of individuals used during the experiment was increased up to 30 individuals.

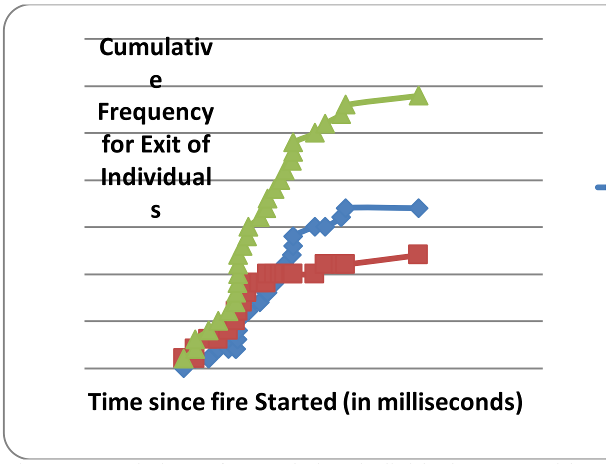

Figure5: Variation of cumulative individual count with time. Scenario 2 (30 individuals), simulation session 1.

During the 1st simulation session for Scenario 2 (Figure 5); 17 individuals have managed to get out from Exit 1, and 12 individuals from Exit 2. This again demonstrates the asymmetrical distribution of individuals among exit points. Clogging/arching effects are observed at both exit points between 60 to 70 seconds (i.e. 60000 to 80000 milliseconds).

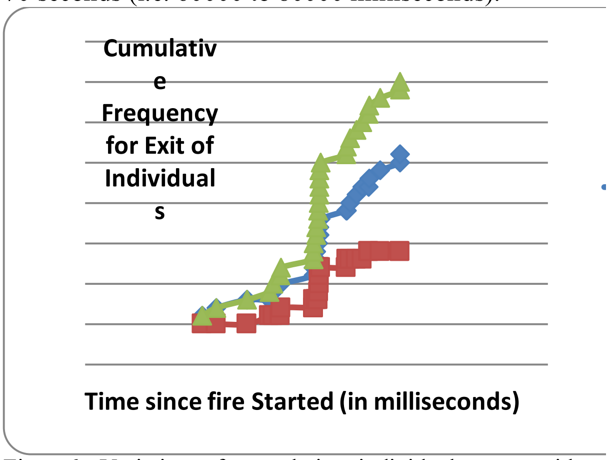

Figure6: Variation of cumulative individual count with time. Scenario 2 (30 individuals), simulation session 2.

Figure 6 shows results for the 2nd simulation session with 30 individuals. During the 2nd execution 21 people have managed to get out from Exit 1 and 9 people Exit 2, demonstrating asymmetry as well as clogging between 70 to 80 seconds (as illustrated in Figure 6). Comparison of results from Figure 5 and Figure 6 reveals that there is a significant difference between the results although they are 2 different simulation sessions related to Scenario 2. However while this difference being evident, still the common observation patterns of crowd behaviour such as asymmetrical distribution of individuals among exits and arching/clogging are preserved.

All the 4 sessions in both Scenario 1 and Scenario 2 (i.e. Figure 3, Figure 4, Figure 5 and Figure 6) reveal that the difference between the 2 graphs in Scenario 2 (i.e. Figure 5 and Figure 6) is considerably greater than the difference evident between the 2 graphs in Scenario 1 (i.e. Figure 3 and Figure 4). This reveals that while our multi-agent based crowd model simulates uncertainty in crowd behaviour during panic; this uncertainty is more evident when the total number of individuals in a Scenario is increased from 10 to 30.

8. Conclusion

The experimental results obtained during evaluation of the prototype indicate the ability of the proposed multi-agent approach to simulate uncertainty of a crowd in a panic. Hence the important aspect learned from multi-agent based approach is its ability to emerge this uncertainty as a result of unplanned knowledge sharing between the individuals.

Unlike the conventional approaches behind developing crowd simulations (such as particle-based approach, fluid-based approach and cellular automata) the agent-based knowledge sharing approach doesn't require developing a global algorithm to simulate the crowd behaviour in groups. Instead with multi-agent based approach assigning a rule-set for each individual with relevant actions for each rule (as discussed during design in Section 5) is adequate.

The initial knowledge of an individual is embedded as a collection of these rules and need not to be complete. Any individual in the simulation can learn something which was previously unknown due to unplanned knowledge sharing with other individuals via communication. Based on this information received through communication (i.e. by message passing) an individual will always update his local ontology (according to Neighbour's Rule Acceptance Criteria in Section 5). Therefore as a result, as time passes each individual will have a better "evolved local ontology" within him/her to decide about the best action(based on Rule Selection Criteria described in Section 5).

Our research work focused about assessing crowd behaviour via developing a computer simulation software and observing a scenario as a simulation. However, another approach is to collect statistical information recorded for a large number of similar scenarios on a given environment setup throughout history. Statistical frameworks [15] can be developed to estimate quantitative metrics such as total number of individuals being escaped in a given time, escape time per individual and percentage of individuals escaped from an exit point. Our crowd simulator can be further calibrated based on these derived statistical metrics by adjusting its configurations. Hence a more reliable level of realism can be guaranteed when results from statistical inference coincide with results from computer simulation.

Acknowledgement

We gratefully thank for all the resources and support provided by Faculty of Information Technology, University of Moratuwa, Sri Lanka.

References

[1] S. R. Musse, and D. Thalman, "A Model of Human Crowd Behavior: Group Inter Relationship and Collision Detection Analysis", in Proceeding of Eurographics Workshop on Computer Animation and Simulations 97, 1997, pp. 39-51.

[2] Cherif Foudil - An Autonomous and Guided Crowd in Panic Situations

[3] Helbing, D.; Farkas, I.; Molnar, P. and Vicsek, T., 2001. "Simulating of Pedestrian Crowds in Normal and Evacuation Situations. In M.Schreckenberg, S.D. Sharma(ed.) Pedestrian and Evacuation Dynamics. Springer Verlag Berlin and Heidelberg, pp. 21-58, 2001.

[4] Helbing, D.; Farkas, I. and Vicsek, T., 2000. "Simulating Dynamical Features of Escape Panic". Nature, v. 407, pp. 487-490, 2000.

[5] Richard Leggette - Real Time Crowd Simulation: A Review

[6] Bruno S. Freya, David A. Savagec, and Benno Torglerb: Interaction of natural survival instincts and internalized social norms exploring the Titanic and Lusitania disasters

[7] Katsuhiro Nishinari, Ken Sugawara, Toshiya Kazama, Andreas Schadschneider and Debashish Chowdhury - Modelling of self-driven particles: foraging ants and pedestrians

[8] A.S. Mikhailova and V. Calenbuhr, From Cells to Societies (Springer, Berlin, 2002).

[9] Katia P. Sycara: Multi-agent Systems

[10] George Rzevski, Petr Skobelev: Emergent Intelligence in Large Scale Multi-Agent Systems

[11] Reynolds. "Flocks, Herds, and Schools: A Distributed Behavioral Model". Proc. SIGGRAPH'87, Computer Graphics, v.21, n.4. July, 1987.

[12] X. Tu and D. Terzopoulos. "Artificial Fishes: Physics, Locomotion, Perception, Behavior". Proc. SIGGRAPH'94, Computer Graphics, July, 1994.

[13] Java Agent DEvelopment (JADE) framework – an Open-Source platform for peer-to-peer Multi Agent-based Applications - http://jade.tilab.com/ (Last Viewed on 29th Jan 2012)

[14] Human Behaviour in Fires – A publication by "Department for Communities and Local Government", UKhttp://www.communities.gov.uk/archived/publications/fire/humanbehaviour (Last Viewed On 23rd June 2012)

[15] [Hiroaki Notake , Yuka Ikehata , Jun'ichi Yamaguchi and Takeyoshi Tanaka - "Fire Risk-Based Evacuation Safety Derived from Statistical Analysis by Type of Building"

[16] MASSIVE – Simulating Life - http://www.massivesoftware.com/ (Last Viewed on 30th June 2012)

[17] Stuart Russell and Peter Norvig - "Artificial Intelligence: A Modern Approach", copyright 1995 Prentice-Hall, Inc.

[18] Ana Luisa Ballinas-Hernández, Angélica Muñoz-Meléndez and Alejandro Rangel-Huerta – Multi-agent System Applied to the Modeling and Simulation of Pedestrian Traffic in Counterflow - Journal of Artificial Societies and Social Simulation 14 (3) 2, 2011 - http://jasss.soc.surrey.ac.uk/14/3/2.html (Last Viewed on 30th June 2012)

Agro-Finance System

Manoj A.Bandara[1], Malith L. Karunaratne[2], D.D.M. Ranasinghe[3]
Department of Electrical and Computer Engineering
Faculty of Engineering Technology,
The Open University of Sri Lanka,
Nawala,Sri Lanka,
[1]asiri.bandara@gmail.com, [2]masthijohna@gmail.com, [3]ddran@ou.ac.lk

Abstract - **Sri Lanka is mainly considered as anagriculture based country, yet the contribution from agriculture to the Gross Domestic Production (GDP)is very low at present. Nevertheless a larger portion of the Sri Lankan labor force is absorbed into agriculture sector. This clearly highlights that there is an issue in productivity in the agriculture sector and this has to be addressed immediately. One major factor that contributes to low productivity is due to lack of knowledge in identifying the most suitable crops to grow in a particular area. This project adopts a Multi – Agent based approach to get information on cultivation and productivity and Neural Network based approach to improve the financial level of farmers. The system is capable of predicting the cost of production as well as the expected price for a selected crop from the productivity enhancement module. The low values of root means square for the difference between the actual price and the predicted price indicate the high accuracy of the system.**

1. Introduction

In Sri Lanka the economy depends to a large extent on agriculture production. The country is predominantly an agriculture based country and agriculture plays a dominant role in the economy in terms of food security, value addition, employment generation and export earnings. The total useable land area is about 1.887 million hectares and out of that about 1.0 million hectares is under permanent plantation crops mainly tea, rubber and coconut. Seasonal crops such as paddy, maize, sugar cane, green gram, vegetables, finger millets cover about 0.887 million hectares. According to 2011 annual report of Central Bank of Sri Lanka agriculture sector in Sri Lanka employs 32.7% of the total labor force of the country in 2010 and 32.9% of the total labor force of the country in 2011.Yet the contribution to the Gross Domestic Production (GDP) is 11.9% in 2010 and 11.2% in 2011. Above statistics shows that engagement of labor force slightly increased and GDP contribution has slightly decreased in the year 2011 compared to 2010. These statistics shows huge amount of labor force engaged in the agriculture sector, but contribution to the GDP is comparably very low. Another considerable factor is that a large amount of the agriculture labor force belongs to rural areas yet they still suffer with poverty withl ow living conditions. Rural poverty in Sri Lanka accounts for 82% of the poor (Department of Census and Statistics, 2008) in the country, with the majority engaged in some form of agriculture. Small farmers in Sri Lanka are unable to engage effectively in agricultural markets which are prone to inefficiencies. The high seasonal, inter and intra-day price volatilities have meant that farmers are unable to plan the type, volume and timing of crop harvest and cultivations to reflect demand conditions.

Another set of major problems that Sri Lankan agricultural community meet are lack of investors, financial support to start and maintain their productivity, transportation, storage and customer interactivity etc. At present dissemination of information and knowledge pertaining to agriculture sector is at a very low level. Because of this reason currently most of the farmers waste their money and crops annually. Currently there are many issues involved in cost of cultivation / production. These costs can be reduced or minimized by introducing an efficient, easy to use method which will give information about all process of production stages to selling process. This will lead to eliminate unnecessary cost of cultivation and will lead to efficient and effective use of resources. Providing correct and timely information to farmers can make them aware of how to enhance the level of productivity to gain a considerable market value to their product.

The proposed approach in this project is mainly divided into two phases. The first phase gives a solution to enhance the productivity of cultivation. The second phase addresses the financial level of the farmers by giving information on production cost and possible market value for their cultivation. Given all these facts the farmer can make a decision on whether to proceed with the selected crop or not.

The productivity enhancement unit is a Multi – Agent based approach where it identifies the best crop to cultivate in a given area. The financial improvement unit adopts a neural Network based approach to give production cost and future market states for the relevant crop.

The rest of the paper is organized as follows. Section 2 gives an overview of existing approaches for similar problems. Section 3 discusses the overview of the design. Section 4 discusses the multi – agent approach to address lack of productivity. Section 5 discusses neural network based approach to financial improvement. Section 6 describes evaluation of the system. Section 7 describes the conclusion of the solution.

2. Existing Approaches

The literature review looks at existing systems in two angles mainly which are developed to provide information on productivity enhancement and on financial improvement.

The study of Premachandra and Ratnayake (2008) propose a knowledge-based approach to land evaluation for the selection of suitable agricultural crops - Crop Advisor. "Crop Advisor" is a Knowledge-based Decision Support System (KBSS) for crop selection. The expert system is powered primarily by human knowledge collected from crop experts. It also considers economic feasibility of raising a crop by taking market price, cost of production, and access to market and yield levels. After consultation with the farmer the system finally suggests a suitable agricultural crop that can be grown in a land unit.

The study by Sami, Habib, Ali (2005), describes grand schemes of a model to be used in an agriculture decision support system. The need of using a combined approach of Multi – Agent System and Constraint programming paradigms is highlighted in this paper. Hence the approach taken is Constraint Programming and Multi-Agent System mixing based on controller agent concept. This system models and simulate the interaction between different actors in the whole process such as negotiations between consumers and water suppliers, and to model decision making process, like the criteria and strategy of water allocation that are used by water suppliers.

Web Based Agricultural Wikipedia and e-Learning System is a project maintained by Audio Visual Centre, Department of Agriculture, Gannoruwa which is a web based Wikipedia and a web based e-learning system (http://www.goviya.lk) developed to facilitate farmer training. The Agricultural Wikipedia contains specific information related to all major local crops and is enriched with audio, video and flash animation. This content will be further developed and expanded with knowledge and information added, edited, changed or deleted on a continuous basis. It is expected that with time it will be developed into a comprehensive compilation of agricultural and agriculture related information.

Nava Goviya is a project carried out by CIC Agrochemicals Ltd[16]. This project focuses on 5 districts – Anuradhapura, Matale, Badulla, Moneragala and Kandy. It seeks to improve agricultural productivity and product quality through a modern online agriculture knowledge learning portal and is developed both in Sinhala and Tamil Languages. This portal carries a comprehensive knowledgebase on different aspects of farming such as pests and fertilizer control, farming management, harvesting, post-harvest technologies, food processing and even on development of farming business skills like market reach, banking and commodity exporting.

MAS are asset of agents which perform communication, coordination and negotiation between each other in order to achieve a common goal. In MAS systems the interactions among agents are autonomous and therefore user intervention is not required for agents to communicate or to carry tasks. MAS provide a coordination mechanism for agents to work towards a common goal.

There are several systems designed to improve financial level of farmers. One of the similar systems in Sri Lanka is Dialoag TradeNet(Dialog TradeNet, 2011). Dialog Tradenet is an innovative solution to overcome information asymmetry in the market, especially for communities at the bottom of the economic pyramid. It provides a multi-model information platform in all three languages Sinhala, Tamil and English and a virtual marketplace - that enables dynamic matching of buyers and sellers. In addition it provides reference prices on demand. Once potential buyers and sellers are matched, the communication is allowed to carry through in different media such as SMS, Voice, USSD and Web.

Another web based system in india is www.agriculturalpriceprediction.com. This website publishes the predicted price data (agricultural price forecasting) of Agricultural commodities for the state of Karnataka in India. Itis using the data provided by AGMARKNET.NIC.IN http://agmarknet.nic.in as a reference source to predict the future prices of Agricultural commodities. Currently, it publishes predictions for two (2) months for thirteen commodities. Agriculture sector involves an immense knowledge contributing from different areas such as environmental conditions, knowledge about pesticides, soil conditions etc. Dissanayake and Karunananda in year 2010have deveopled a multi agent system where the agents contain the domain knowledge of various aspects related to agriculture [28]. When pose a problem these agents are capable to communicate and negotiate with each other and come up with a reasonable answer as humans do. The prototype system was build to answer questions for growing green chillie.

The above reviewed systems mainly provide information stored in databases as per user requirements. Therefore those systems lack the ability of considering current changes and lack the ability of future predictions.

3. Design of the System

As stated earlier the productivity enhancement module and the financial improvement module are implemented as two separate modules of the system.

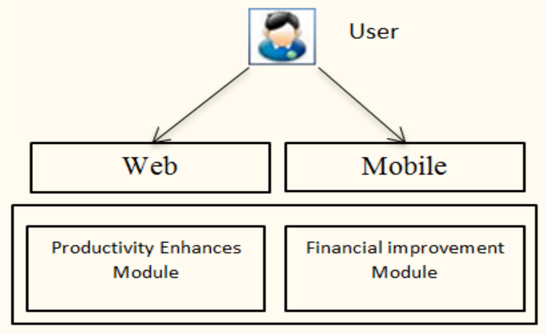

Figure1: Overview of the system

Figure 1 shows overview of the design. The productivity enhancement module and the financial improvement module connect the user through a web application and a mobile application. The productivity enhancement module is designedusing agent technology and providesinformation to select the best crop for cultivation. The financial module developed using a neural network gives the cost associated with the selected crop. Since the farmer is equipped with the knowledge of the suitable crop for cultivation, steps that should follow to get the maximum harvest, cost and profit to relevant crop, the farmer can make a better decision on the path he has to follow.

A. Productivity EnhancementModule

The productivity enhancement is mainlydivided into five stages as shown in figure 2.

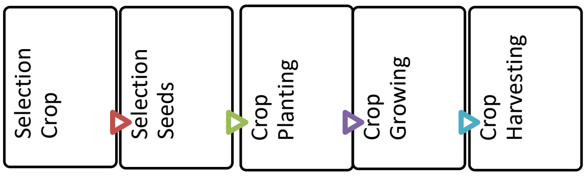

Figure2: Process of Productivity Enhancement Module

The first stage is the Selection of themost suitable Crop for cultivation according to given environment conditions by the farmer. The second stage is Selection of Seeds which identifies the most suitable type of seed for the selected crop type in stage1.Third stage is Crop planting. This stage gives information about methods of planting techniques and method of using Land for crops. Fourth stage is about Crop Growing which provides details about what type of fertilizer to use, fertilizer plan, controlling techniques and management of pest and diseases. Fifth stage is Crop Harvesting, which provides information about techniques and details of harvesting and storage techniques.

Productivity enhancement module goes through the overall process of agriculture providing accurate, timely and actionable decision making information right from selecting a crop to growing and to harvesting. To achieve this goal this module is being created using multi agent technology. Multi – agent system (MAS) is used in this project to handle the complexity of handling information in a parallel manner.

B. Financial Improvement Module

This Financial Improvement Module is designed to predict production cost, future market vlaues and climate condition such as rainfall for a selected crop from the productivity enhancement module.

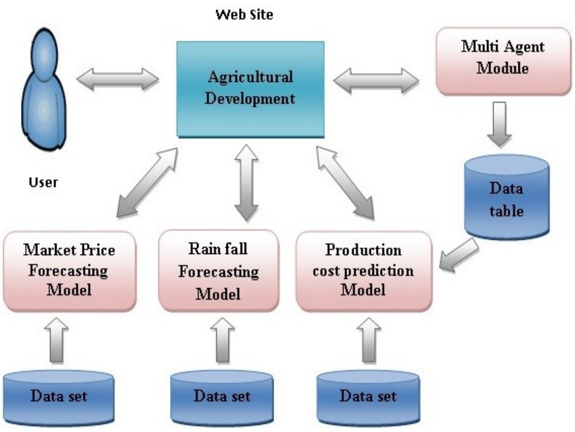

Figure3: Financial Improvement Module

There are three main models in this module.

a. Market Price Prediction Model
b. Rainfall Prediction Model
c. Production Cost Prediction Model

The goal of this module is to provide predicted future market prices and rainfall values from which farmers can know how the prices will change in the future. By giving a predicted production cost farmers can take decisions to choose which crop to cultivate and how much cost will have to bear. This prediction will give an opportunity to the farmer to take an intelligent decision with respect to financial situations in cultivating a particular product.

C. User Interaction

User can interact with the system mainly through the web application. User does not need to log-into the system to access the service. Hence the system is open to any farmer who needs information with regard to production enhancement and financial improvement.

4. Multi–Agent Based Productivity Enhancement Module

Productivity enhancement module mainly consists of three layers as shown in Figure 4. The layers are User Application Layer, Communication Layer and Multi agent Layer. Multi – agent layer consist of agent swarms and ontologies. These agent swarms work along with ontologies to provide decisions to farmers and the generated decisions are send back to the user through the communication layer.

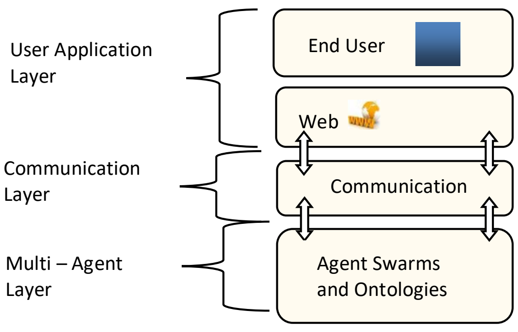

Figure4: Layers of productivity enhancement module

One role performed by the communication layer is transferring information to the user application layer coming from multi–agent layer. The other role is, passing the user request to the multi–agent layer coming from user application layer. User application layer consist of web applications and allow users to interact with the system more easily.

An intelligent behavior is added to the productivity enhancement module through agent swarms. Ontology provides the foundation intelligence for behavior of agent swarms. This module contains six resource agent swarms and one Admin agent swarm.

Figure 5shows top level architecture of productivity enhancement module. User requests are taken to the system using web application and are send to central agents swarm. Admin agents swarm keep user request in the central message space. Then other six resource agents check whether the request belongs to them or not. If request belongs to one of the resource agent swarm then accept it and process it and after the work is done the result is send back to the central message space. Then that result accepts from admin agents swarm sends it to the web service. After that web service send the result set to the web application and the output is given to the user. The results are saved in the database attached with the web application.

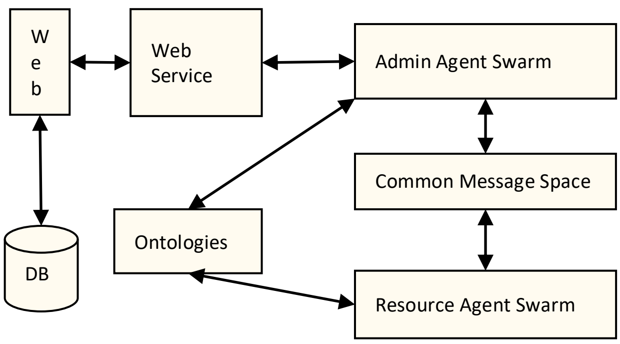

Figure5: Productivity Enhancement Module

Admin Agent Swarm contains four agents. One agent in the swarm administrates the admin agent swarm. This agent creates all other agents when system initializes. Another agent in this swarm handles the communication between admin agent swarm and resource agent swarm. Other two agents handle web service request coming into the productivity enhancement module and responses sends by productivity enhancement module. Common message space is the hub for communication between Admin Agent swarm and Resource Agents swarms. All agent swarms are able to write messages to the central message space hence other agent swarms are able to respond to those messages.

There are six resource agent swarms available in this system namely, Crop Agent Swarm, Seed Agent Swarm, Plant Agent Swarm, Grow Agent Swarm and Harvest Agent Swarm. Figure 6 shows Resource agent swarm architecture. It contains request agent, central admin agent, and set of end point agents, message space and specific ontology for a given knowledge base to the swarm. Response to the request agent's message is written on the central message space. The response from central admin agent is send to the swarm message space. After message is written to swarm message space, end point agents access it and respond to it. Central Admin Agent negotiate with the set result and send correct result out from the swarm using request agent which writes on the central message space. End point agents exit from the system at the end of their job.

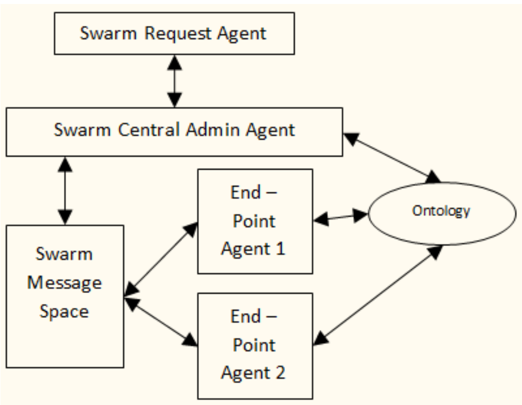

Figure6: Resource Agents Swarm Architecture

All six resource agents swarm contain their own ontology which is used to store the knowledge required for the execution of agents and to enable interaction among each other. The six ontologies identified from the productivity enhancement module contain strong domain knowledge related to agriculture. In addition to the six resource agent swarm ontologies the AgriInfo main ontology module too is included here. This ontology work with Central Admin agent swarm and it contains all the knowledge required to process the system.

If user request to identify the most suitable crop for a given environment that request is placed on the central message space. Then crop agent swarm access that request through request agent in the swarm. After it takes from the central admin agent and send it to the message space. When request comes to the resource swarm End point agents are created and access the message space. Crop ontology used to gain domain

knowledge to process the required action to the end point agents. After each agent finish their work send the result to the swarm message space and central admin negotiate with results. Final outcome sends to the central message space through the request agent.
All the decisions taken by farmer are stored in the Database attached with web application. The financial improvement module can access those data for further processing of the system.

A. Implementation

Productivity enhancement module is implemented using Multi Agent based technology. Agents swarms are created using JADE framework which compiles with FIPA standards. The overall multi agent layer developed using J2SE platform. Protégé framework was used to model the ontology with the necessary knowledge regarding different domains of agriculture. Communication layer created following SOAP web service. User application layer is designed using .NET framework 4 and ASP.NET. To create user application layer database used SQL Server 2008 R2.

5. Neural Network Based Financial Improvement Module

Second module of this system is financial improvement module which is a Neural Network based approach and it predicts production cost,future market values and future rainfall values for relevant crops.

A. Market Price Prediction Model (MPP model)

The Market Price prediction model includes a feed forward multilayer neural network which predicts the market prices for selected crops. A data set is inserted to train the neural network which consists of all island retail prices of the selected cropson daily/weekly/monthly basis for the last five years. These data are then normalized and preprocessed to prepare them for the model building process to achieve an efficient and effective forecasting process.
For each crop, a separate data set is prepared and used to predict the market price of that crop. For all the selected crops each input data set includes;

- Week/Day/Month of the Year
- Price

The first column (day/week/month of the year) is selected as the input and second column (market price) is selected as the output and as the value to be predicted.

B. Rainfall Prediction Model (RP model)

The rainfall data are collected on daily/weekly/monthly basis for the past five years and they are used to predict future values. This also includes a feed forward multilayer neural network.
Separate datasets consisting of daily/weekly/monthly rainfall data are used as inputs and each input dataset includes,

- Week/Day/Month of the Year
- Rainfall

Same method which is used to forecast market price is applied here to forecast the upcoming rainfall values.

C. Production Cost Prediction Model(PCP model)

This model also includes a feed forward multilayer neural network which predicts the production costs for the entries output by the Multi agent module. These entries include data on crop type, seed type, land size, planting technique, type of fertilizer and harvesting technique. These data are stored in a data table which is accessed by the Production cost prediction Model.

Table1. Data set to train PCP model

Input						Output
1	**2**	**3**	**4**	**..**	**10**	11
65.48	**68.72**	**70.26**	**71.25**	**..**	**100.26**	?
68.72	**70.26**	**71.25**	**75.83**	**..**	**98.36**	?

A separate data set is inserted to train the neural network which consists of data such as crop type, seed type, land size, planting technique, type of fertilizer and harvesting technique and production cost. The last column is used as the output and all the other columns are used as the inputs to the neural network. These data are collected from the Department of Agriculture. The data stored in the table which are taken out from the multi-agent module, are used to do the predictions.

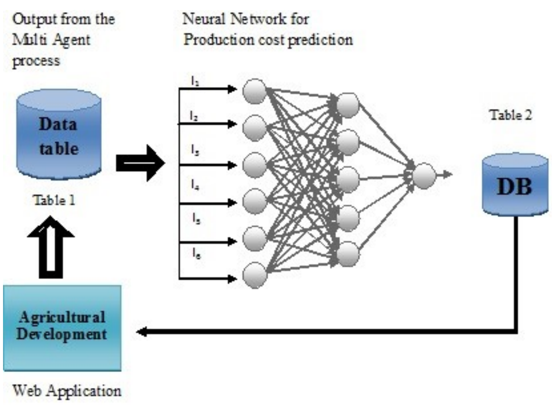

Figure7: Overview of Production cost prediction Model

The method used for market price and rainfall prediction is called windowing. The basic idea is that data set is divided into two windows (groups) of fixed sizes which contain input data and target data. The data in the input window is inserted to the neural network as inputs and the data in the target window is used as the output value. The value in the output window is used to compare with the predicted value of the neural network and then the neural network is further trained to reduce the minimum difference between the predicted value and the actual output value. This process of training is carried out step by step shifting both input and target windows down along the data set while reducing the difference.

Table2. Data input method for Price Prediction

Crop Type	Seed Type	Planting	Land size	Fertilizer type	Harvesting	Production cost
Tomato	S11	PT1	4.5	F11	H11	70,000
Potato	S21	PT1	6	F21	H12	86,000
Onion	S32	PT3	4	F31	H11	64,000

A window of 10inputs (10 consecutive prices) is fed to the neural network and the network will try to predict the 11^{th} value, corresponding to the next week in the row, of each of the crop.

D. Training

Training is the process of adjusting weights and threshold values of a neural network to get the desirable outputs. The training method used here is the Resilient Propagation training. The resilient propagation training (RPROP) algorithm is usually one of the most efficient training algorithms provided for supervised feed forward neural networks. One particular advantage to the RPROP algorithm is that it requires no setting of parameters before using it.There are no learning rates, momentum values or update constants that need to be determined. This is one of the best methods because it can be difficult to determine the exact learning rate that might be optimal.

E. Data Normalization

To get the best performance from the Neural Network each input variable should be preprocessed and normalized so that themean value, averaged over the entire training set, is close to zero, or else it is small compared to its standard deviation. The ranges of the indexes vary slightly, as their domain is totally different. The purpose of normalization is to modify the output levels to a reasonable value. Without such transformation, the value of the output may be too large for the network to handle, especially when several layers of nodes in the NN are involved.
A transformation can occur at the output of each node, or it can be performed at the final output of the network. In normalization original values are mapped to a smaller range of value set (-1, 1 or 0, 1).In order to normalize each index range to [0, 1], the following simple formula is used:

$$\text{Price}(x) = (Px - P_{Min}) / (P_{Max} - P_{Min}) \quad (1)$$

Price(x) = New value after scale
Px = Actual value before scale
P_{Max} = Maximum value
P_{Min} = Minimum value

Thus each of the input variables will lie in the same range (0, 1).

F. Selection of Optimum Neural Network

Several techniques were used to implement the program and to improve the forecasting performance of the neural network. Initially, Group of neural networks is created having different combination of hidden layer processing units and learning rates. Finally the neural network which has a higher performance of forecasting according to the performance function is selected.
Neural network training algorithm was designed to train the neural network and the same time simulate the neural network with testing data set and calculate the root mean square(RMSE) error using two value sets simulate output vector and target actual data vector. The training error is used to select the best suitable network by getting the error value for given number of iterations.

Error = ideal value – actual value(2)

RMSE = (ideal value – actual value) / Set size(3)

This evaluation was done for all neural networks. Then the neural network structure having optimum forecasting performance was selected based on the RMSE value.

6. Evaluation

Several testing methods were used to evaluate the productivity enhancement module. At the development level each module was subjected to unit testing and integration to ensure proper functionality of the module. After integrating the individual modules the entire system was subjected to system testing. Evaluation has been done in terms of accuracy of processing and final decision respect to different information types using questionnaires where agriculture field experts participated in the evaluation process.

Agent Module	Output Decisions	Accuracy
Selection Crop	100%	88%
Selection Seed	98%	85%
Crop Planting	95%	83%
Crop Growing	96%	80%
Crop Harvesting	94%	76%

Table3: accuracy level of productivity enhancement module

According to above evaluation results it is evident that the productivity enhancement module has a high accuracy.

System has been tested on real data: daily vegetable market prices are collected for the period 01/01/2007 to 15/03/2012. There are 250 data points without any null values.
Data from 01-01-2007 to 22-12-2011 were used as the training set and from 01-01-2012 to 08-03-2012

period was used for testing purposes. Following table displays the predicted prices for the potato crop.

Table 4: Predicted results for potato

Date	Actual Price	Predicted Price	RMS Error
2012-01-01	127.33	127.62	0.0038
2012-01-08	128.5	128.42	0.0011
2012-01-15	124.17	124.29	0.0016
2012-01-22	136.5	136.23	0.0036
2012-02-01	130.43	130.37	0.0007
2012-02-08	122.61	122.67	0.0008
2012-02-15	116.84	116.85	0.0002
2012-02-22	114.5	117.16	0.0347

As seen in the above table, for a given date/week the predicted price is calculated and the rms error between the actual price and the predicted price is also calculated.

7. Conclusion

The proposed system includes two main modules,productivity enhancement module and the financial improvement module.The productivity enhancement module is capable of identifying the most suitable crop to grow in a given land area. The evaluation shows that the system is capable of predicting the crop type with a high accuracy. Due to very low RMS error it is evident that the prices predicted are comparable with the actual values. Therefore we can conclude that with the implementation of this AgroFinance system we have been able to address the problem in the gap of making an intelligent decision on the type of the crop to grow based on financial situation.

Acknowledgement

We would like to thank Dr. RohanWijekoon, Deputy Director Department of Agriculture and Mr. Sisira Kumara, Assistant Director of Agriculture, Audio Visual Centre, Gannoruwa who helped us by providing the necessary data and instructions on various agricultural sections.

References

[1] Bowerman, B. L., and O‟Connell R. T. (1987). Time Series Forecasting,Unified Concepts and Computer Implementation, 2nd edn. Duxbury Press, USA.

[2] Feng Lin, Xing Huo Yu, Shirley Gregor, and Richard Irons. (1995).Time Series Forecasting with Neural Networks. Department of Mathematics and Computing ,Central Queensland University , Australia.

[3] Dialog TradeNet [Online]
Available :http://tradenet.dialog.lk/

[4] Igor Rubanov, and Vladimir Rubanov.(2005). Using Neural Network for Time Series Forecasting.8th International Conference on Pattern Recognition and Information Processing,Minsk,Belarus.

[5] SriganeshLokanathan , Harsha de Silva , Iran Fernando,"Price Transparency in Agricultural Produce Markets" Sri Lanka –Draft 2008

[6] Department of Census and Statistics, Agriculture and Environment Statistics Division, [Online] Available at: http://www.statistics.gov.lk/agriculture/index.htm

[7] Central Bank of Sri Lanka. (2011). Annual Report.

[8] De Silva, H. ,Ratnadiwakara, D.,(2008).Using ICT to reduce transaction costs in agriculture through better communication: A case-study from Sri Lanka. Canada: International Development Research Centre.

[9] Premachandra, M. M. A. ,Ratnayake, U. , (2008). , Advising Farmers on Crop Selecting Using Expert Systems: Proceeding of the fifth Annual Sessions. Colombo: Sri Lanka Association for Artificial Intelligence (SLAAI).

[10] Prasad, G.N.R. and Babu, A.V. (2006)., A study on various expert systems in agriculture. Georgian Electronic Scientific Journal: Computer Science and Telecommunications.

[11] Chakrabarti DK, Chakraborty P. (2008) An Example of Agricultural Expert Systems Being Used in India. Georgian Electronic Scientific Journal: Computer Science and Telecommunications 2008;No.1(15).

[12] Chakrabarti DK, Chakraborty P. (2008) A brief survey of computerized expert systems for crop protection being used in India, ScienceDirect: progress in Natural Science 18, 469-473.

[13] Dissanayake, D. M. S. ,Karunarathane, A. S.,(2010), Investigating Use of Multi- Agent Technology for Agriculture Information Security System. Colombo: Second joint national conference on information and technology in agriculture.

[14] Sami, A., Habib, A., Ali, N. , (2005), Constraint Programming and Multi-Agent System Mixing Approach for Agricultural Decision Support System, Emergent Properties in Natural and Artificial Dynamical System in ECCS'O5 International Conference, PP. 113-119.

[15] Web Based Agricultural Wikipedia and e-Learning System, Audio Visual Centre, Department of Agriculture, Sri Lanka. [Online]. Available: http://goviya.lk/

[16] Nava Goviya, CIC Agrochemicals Ltd. [Online]. Available: http://navagoviya.org

[17] Java Socket Programming, Oracle Corporation. [Online]. Available:

http://docs.oracle.com/javase/tutorial/networking/sockets/index.html

[18] Jade Documentation, Jade Community. [Online]. Available: http://jade.tilab.com/doc/

[19] The Protege, Stanford Center for Biomedical Informatics Research. [Online]. Available: http://protege.stanford.edu/

[20] Jade Tutorial, Jade Community. [Online]. Available: http://jade.tilab.com/doc/tutorials/JADEAdmin/startJade.html

[21] Jade Tutorial, University of montreal. [Online]. Available: http://www.iro.umontreal.ca/~vaucher/Agents/Jade/primer2.html

[22] SOAP Web Service, Developer Blog. [Online]. Available: http://blog.jdevelop.eu/2008/02/27/creating-soap-web-services-with-netbeans-6/

[23] ASP.net web forms, Microsoft Corporation. [Online]. Available: http://www.asp.net/web-forms

[24] Introduction to web service, Oracle Corporation. [Online].Available: http://netbeans.org/kb/docs/websvc/intro-ws.html

[25] Getting started with JAX – WS web services, Oracle Corporation. [Online]. Available: http://netbeans.org/kb/docs/websvc/jax-ws.html

[26] Jade Forum, Telecom Lab ITALIA. [Online]. Available: http://sharon.cselt.it/pipermail/jade-develop/2002q1/000179.html

[27] Protégé Wiki, Stanford Center for Biomedical Informatics Research. [Online]. Available: http://protegewiki.stanford.edu/wiki/Protege4GettingStarted

Agent Negotiations for Improving Quality of Solutions from Multiple Perspectives

M.D.W Srimal[1], Asoka S Karunananda[2]
Faculty of Information Technology, University of Moratuwa
Katubadda, Sri Lanka
[1]woshan.srimal@gmail.com, [2]asoka@itfac.mrt.ac.lk

Abstract -**Most of the real world problems can be solved using more than one method which may return slightly different solutions. For instance, statistical techniques, artificial neural networks, fuzzy logic and genetic algorithm can model the same real world problem subject to own strengths and weaknesses. However, it is evident that human beings can modify/improve solutions generated in the individual capacity through negotiations among the individuals. This concept has been employed in the Multi Agent Systems (MAS) technology which can model complex real world problems to achieve quality solutions beyond the individual capacity. In this work, MAS has been used to ensemble weather forecasting results individually generated by Artificial Neural Network (ANN) and Genetic Algorithms (GA) through negotiation among solutions. It considers ANN and GA as two agents. It has selected this application domain to demonstrate the concept since weather forecasting is important for many sectors such as agriculture, fisheries and transportation. Our MAS solution forecasts the rainfall for next twenty four hours with the use of set of present weather conditions as inputs for ANN and GA agents. The defined two agents are used to operate on an ANN and GA solutions that start negotiation & deliberation to produce a more rational forecasting. The experiment concludes that even when solutions by ANN Agent and GA Agent shows a disparity at the beginning, they reach to commonly agreeable solution through the negotiation in the multi agent solution with a 65% of success.**

1. Introduction

A rational problem solving strategy has a great importance in any domain such as Science, Engineering, Biology, Commerce and Arts. Most of problems in these domains can be solved using more than one method which may return slightly different solutions. The common approach has been used a single technology or method to solve a particular problem. Yet, these technologies have their own strengths and weaknesses unique to them which directly affect the qualities of the final result. By using multiple technologies to solve a single problem, the limitations of the individual technologies can be controlled while enabling maximum use of their strengths in to the final solution. This concept has been employed in the MAS technology which can model complex real world problems to achieve quality solutions beyond the individual capacity. In order to show the concept, the multi agent negotiations based multiple perspective approach was used in the weather forecasting domain since an accurate weather forecasting mechanism is an essential requirement in any country. Any accurate weather forecast can lead to the mitigation of risks associated with sectors like Agriculture, Transport, Fishing, etc. Governments and other organizations also use weather forecast reports to protect life and properties [1].

Recently weather forecasting model development efforts have been made to obtain better forecast models by using ANN, Fuzzy Logic (FL) and GA kind of none algorithmic problem solving techniques available in the field of Artificial Intelligence. Different types of ANNs have been proposed to forecast weather results. Back Propagation Neural Network (BPNN) and Radial Basis Function Neural Network (RBFNN) has been used to efficiently generate forecast results [2], but final outcomes were biased to the trained data set and the false positive rate was around thirty percent. ANN approach also can be used for real time weather forecasting [3] by using data from all combinations of atmospheric conditions to train the network for limiting the biasness of the results. An ensemble of ANN techniques like Multi-layered Perception Network (MLPN), Elman Recurrent Neural Network (ERNN), Radial Basis Function Network (RBFN) and Hopfield Model (HFM) has produced accurate forecast results [4]. The problem of less accuracy in forecast reports has been addressed using Dynamic Weighted Time Delay Neural Networks (DWTDNN). Its main focus has been to forecast both temperature and rainfall [5] by treating weather forecasting as an unbiased time series forecasting problem. GA based forecasting approach have also been used for weather forecasting in order to filter out sets of historical data for K-Nearest Neighbour (K-NN) model. K-NN model then classify the weather report in an appropriate way to generate accurate results [6], but this method is a time and resource consuming method. Further, GA has been used to populate mathematical models to predict the rainfall over India that best describe the temporal variations of the seasonal rainfall and therefore enables the forecasting of the future rainfall [7]. Although there are number of mathematical models that can be created using this method, only the particular region will be represented in the models. Therefore forecasts for other regions may not be accurate. Combined FL and K-NN are also used to predict weather conditions [8]. K-NN is used to filter the fuzzy membership functions. Richness of the training dataset determines the quality of the results generated from the fuzzy logic system. Although there are number of non algorithmic forecasting models available, false positive rate of forecast reports seems to be considerably high.

In order to solve the false positive rate of the forecast results problem, multi agent negotiation based ensemble of multiple perspective approach was proposed through this work. Meteorological experts and related staff can be considered as the users of the proposed system. After giving present weather conditions such as Maximum temperature, Minimum temperature, Relative humidity day time and Relative humidity night time, Rainfall as inputs to the system, the rainfall forecast for next twenty four hours will be produced. Two agents were assigned to generate rainfall forecasting reports independently by using ANN and GA based forecasting models. In order to release the conflicts among agents, the negotiation and deliberation process is used. Java Agent Development Environment (JADE) is used to develop the Multi agent negotiations architecture. The ANN and GA toolkits available in MATLAB are used to develop weather forecasting models. The historical weather data set related to the Colombo city was taken from department of meteorology Sri Lanka. It includes the weather data from 1st of October 2011 to 30th of November 2011. This data were used to train and develop the ANN and GA based forecasting models. Further, part of the dataset was used to test the system. Finally, in order to show that the proposed concept is functional, the negotiated rainfall forecast results have been plotted which shows that they reach a commonly agreeable solution through negotiation. This system has a set of capabilities such as the ability to work with incomplete and noisy data. It uses both data and expert knowledge to produce results. Only disadvantage is that it takes considerable amount of time to produce results due to negotiation process.

The rest of the report has been organized as follows. Section 2 reports on critical and organized analytical reviews on set of works in the weather forecasting domain. Section 3 reports on the relevant technologies. Section 4 depicts an approach to multi agent negations for weather forecasting. Section 5 gives the design of the negotiation model. Section 6 demonstrates the implementation of the negotiation model. Section 7 discusses the evaluation and Section 8 gives the conclusion and further work.

2. Existing Weather Forecasting Models

When considering the domain of weather forecasting, various types of ANNs, GA based models and FL based combined models have been widely used to come up with better forecasts. Thus it is evident that a single problem can be solved using different technologies. Examples of these models are presented throughout this section.

A. Artificial Neural Network Based Models

Tiruvenkadam & Subhajini, [2] proposed and developed an artificial neural network (ANN) based weather forecasting system by highlighting the strengths of ANN like non algorithmic problem solving techniques with indications of their simplicity and robustness features. This study has been used in the Back Propagation Neural Network (BPNN) and Radial Basis Function Neural network (RBFNN) to generate weather forecasts. As training data they have used a sample real dataset of a particular area from a metrological department. The factors temperature, air pressure, humidity, cloudiness, precipitation, wind direction and wind speed of weather forecasting are used from meteorological experts to train the models. Finally they have come up with a rainfall prediction system. Through the study, the performance of BPN and RBF is also compared and stated that RBF is much accurate and efficient. By looking at the solution it can be seen that the final results are depend on the trained data set and it generates more biased results to the trained data set. When looking at the available statistics, false positive rate of forecasting were around thirty present.

Hung et al, [3] have presented a real time rain fall prediction system by using set of ANN techniques while improving the performance of forecasting. Study was based on the Bangkok City, Thailand and the entire research was based on collected weather data from seventy five locations in the city around the predicting point for four years of time. Entire data set had been used for developing ANN model and authors have found that the model based on feed forward ANN model using hyperbolic tangent transfer function achieved the best rainfall prediction than other models. Work has used a combination of parameters such as relative humidity, air pressure, wet bulb temperature and cloudiness etc to form the model from surrounded data collection stations and allow producing the rain fall reports in real time manner. The authors have used different copies of ANN model to train them using different datasets (from rainy and none rainy periods) to achieve unbiased forecasting results. Finally the author has produced six types of rainfall prediction reports ranging from one hour to six hours ahead. Among them according to the author one hour to three hour results were more accurate than others. Final results were generated using a single output from the model and there was a probability to generate false positive results due to the independency in decision making of the model.

Maqsood et al, [4] proposed somewhat different approach for next twenty four hour weather forecasting by using ANN and associated technologies. Their approach consists of an ensemble of seven different neural network models. Each ANN model uses separate technologies (Multi-layered perception network (MLPN), Elman recurrent neural network (ERNN), radial basis function network (RBFN), Hopfield model (HFM), predictive models and regression techniques) to perform the prediction and final prediction report comes as a collective answer from all the models. In assembling process, they have proposed a weighted assembling method depending on the ANN model's error rate. Each model has been separately trained with a common data set, which includes Temperature, Wind speed and Humidity. Once an independent forecast was generated from each model, all the outputs go through a module called average results generator. Finally the entire system produced an average forecast report for a specific time. By looking at the approach, it can be seen that the final result comes just as an average value. Rather doing this way to get the final result, if it is possible to go to negotiation based process among each model to conclude the final answer, it would have been more rational and of less false positive results.

S. Santhosh and Sheref, [9] has proposed a neural network based algorithm for predicting the temperature. According to the authors, the major problems of weather predictions are the error involved in the measurement of initial conditions in atmosphere and incomplete understanding of the atmospheric process. Due to above problems the traditional weather forecast results become less accurate. In order to solve such problems they have proposed a temperature prediction system using Back Propagation Neural network. This network contains three layers called input layer, hidden layer and output layer. For training of the system they have used a real time weather dataset that contains about two hundred entries. According to the approach it is a simple and efficient method for making any data related predictions but they have concluded that BPN as the better method available to make this kind of predictions, without testing other available Neural Network models.

James N,K.Liu and Raymond S.T.Lee [10] has proposed a methodology to short term rain fall forecasting in a relatively small region using a Back Propagation Neural Network model. Data sets from eleven weather stations were used for training the model. Six hourly data of dry bulb and wet bulb temperatures, rain fall, mean sea level pressure, relative humidity, wind direction and wind speed were used from selected weather stations to train the model. In order to solve the incomplete data problem, linear function interpolation with nearby value mechanism has been used. The proposed method has included a GA with Elitism mechanism for selecting the high fit data to train the ANN from transformed data set. Fuzzy based classified rainfall parameters were used as the expected output from neural network when training the model using BPN mechanism. Finally the model has been tested by using three data sets. They are Set A: Weather data from single station at Hong Kong, Set B: Weather data from multiple stations at Hong Kong, Set C: Time series weather data from multiple stations. Then linear correlation and average classification method is used for analyzing the model and according to the outcome of them, authors have concluded that the multiple station model is more accurately when predicting the rainfall over other models. According to the work done, the accuracy of model depends on richness of the data set that is used for model training.

L.L LAI et al [5] proposed an intelligent weather forecasting model. Normal difficulty in simulating the meteorological phenomena and the corresponding characteristic of weather forecasting has been addressed in the work. By using complex differential equations and computational algorithms it is difficult due to none linear nature of atmosphere when forecast the conditions. Through this work authors have proposed a methodology to forecast short term temperature and rainfall. In order to do so they have proposed Dynamic Weighted Time Delay Neural Network (DWTDNN) with suitable data preprocessing mechanisms. Authors have also mentioned the strengths of ANN technology to forecast on various areas such as short term load forecasting, agriculture risk management. Due to difficulty in capturing time related dynamics of time series events authors have mentioned that the existing multi layer neural networks were not suitable for weather forecasting like time series predictions. DWTDNNs are kind of dynamic neural networks which maintain a short term memory mechanism to store dynamic data. In order to train the proposed network they use pre processed data from fourteen meteorological centers around shanghai. According to their test results, the proposed system has given that approximate weather results. Although it gives approximate results, there are improvements to be placed in the architecture because still the false positive rate can be seen in the final results as shown in drafts.

B. Genetic Algorithm Based Models

Yuen Keith, [4] proposed an approach to rain fall prediction system by using hybridized genetic algorithm (GA) and K-nearest neighbour (K-NN) classifier based on a set of weather data. Normally it is difficult to perform a classification on huge amount of data to get proper predictions on rain fall, but through this work, the author has used a genetic algorithm with a fitness function to select highly fit data for the rain fall classification process based on K-NN algorithm. Further the author has done a set of experiments and proved that the accuracy of the heavy rainfall prediction processes. According to the work done by the author, the approach is suitable only for heavy rain fall forecasting and time taken to generate high fit data is considerable.

C. M. Kishtawal et al [8] have proposed a genetic algorithm based rainfall forecasting model. Through the study they have assessed the feasibility of a nonlinear technique based on a genetic algorithm for the prediction of summer rainfall over India. The Genetic algorithm has been used to populate mathematical models to predict the rainfall. The genetic algorithm finds the equations that best describe the temporal variations of the seasonal rainfall over India, and therefore enables the forecasting of future rainfall. The forecast equation developed in this study uses the monthly mean rainfall during June, July, and August for the past years over five rainfall homogeneous zones of India to predict the seasonal rainfall over the Indian landmass. They have applied the GA to find the equation that best fits the rainfall data in one part of the dataset called the training set. The predictability skill of the solution equation then has been validated using a separate dataset. Authors have attempted to pick an equation in such a way that the difference between the fitted and the actual values of the time series does not exceed a threshold at any instance, in addition to the fact that such equation should achieve the maximum possible fitness strength. The major advantage of using genetic algorithm versus other nonlinear forecasting techniques such as neural networks is that GA produces a model as an equation for rainfall forecasting. It is to be noted that due to the limitation on the population of candidate equations, or the number of operators allowed in an equation, it is difficult to obtain the most appropriate expression at the end of the initial evolution process. Although there are number of mathematical models that can be created using this method only a particular region will be represented in the models. Therefore forecasts for other regions may not be accurate.

Juan Peralta et al [11] have introduced the comparative method for time series forecasting. This is important for displaying the manner in which the past continuous conditions affect the future conditions. Authors have used the concept of evolving neural network to forecast the weather like conditions. They have comparatively used two neural network evolving methods to develop the system, where one was carried out with a genetic algorithm and the other with an estimation of distribution algorithms. The main objective of this study is to; take an attempt to improve the accuracy of final forecasting results. The genetic algorithm based model has been used to define best neural network model. This process has been involved to define the best fit neural network properties such as number of hidden layers, activation functions etc... Authors have used five types of time series data for experiment setup and finally have concluded the Strength of GA based forecasting model.

C. Fuzzy Logic Based Combined Models

K. Hansen & Denis Riordan, [8] have proposed a FL based weather forecasting system for an airport area. According to the problem they have addressed, it is a small area prediction within small time periods when compared to the traditional weather forecasting problems. They have proposed a fuzzy logic and K-NN combined method to solve their problem. They have used K-NN algorithm to produce, fuzzy membership functions required to form a FL model from thirty six years of collected data set. Here they have worked with traditional weather prediction experts to tune the FL system to produce accurate results. Finally by testing this system with existing systems, and it is showed that the accuracy of their system. By looking at their work they have addressed the problem of retrieving required fuzzy membership functions for the fuzzy logic model and implementation of FL system.

Raymond Lee and James Liu, [12] have proposed an intelligent weather forecasting system. Main objectives of their work was to reduce the computation power require to generate weather forecast results compared to traditional methods and generate forecast results within short periods of time for relatively small regions. In order to achieve these objectives they have used Intelligent Java Agent Development Environment (iJADE) to develop Fuzzy Neuro based intelligent agents for automatic weather information gathering and filtering and for time series weather prediction. The iJADE provides an intelligent layer which supports to develop intelligent capabilities other than communication and mobility in JADE and IBM Aglets. According to the proposed system it contains five major components as follows. User Requirement Definition Scheme for collection of specification of the expected report. The Weather Reporting Scheme is for handling final reports. Data collection scheme is used as a set of mobile agents to visit remote weather stations and collect weather data. Variable selection and Transformation scheme selects appropriate weather data to train and test the system. Fuzzy Neuro Training and Prediction Scheme are used for training and making predictions according to the data sets from a variable selection scheme. When looking at the test result provided by authors it can be concluded that the proposed method is better than the single station prediction model.

Predictions in Financial, Production, Climatology, Traffic control etc can be taken as time series prediction problems. There are mathematical models available to make predictions for those areas. However those models require an accurate set of initial input values for models. Otherwise small errors in initial input values lead to inaccurate predictions. Practically having an initial input values is impossible. In order to overcome the above problem Luis et al, [13] have proposed the Adaptive Network-based Fuzzy Inference System (ANFIS) and a Genetic Algorithm based two approaches with comparisons of each other. As the data set for construct the two models they have used microscopic traffic of an automobile through a succession of two hundred and nine traffic light positions. Dataset was used in fuzzy model training purpose and construction of fuzzy membership functions in ANFIS model. GA model is also used on the same dataset to generate the set of rules associated with predictions. Only the highest fit rules were chosen for use. Finally authors have compared both models and stated that the ANFIS model works better for predictions compared to GA model. Error rate, time consumption and processing were lower in the ANFIS model compared to GA model.

Through the literature survey discussed above, the three major Artificial intelligence approaches (ANN, GA, FL) were widely used by authors to solve the problems in weather forecasting. Among those approaches more than eighty percent of researches have been conducted using ANN and limited amount of researches have been conducted using GA, and FL. High false positive rate of reports, being biased to the trained dataset, taking more time to produce results (cannot be used for real time predictions) and models are not generalized to the areas are the identified problems of the three technologies through the survey. By analyzing above problems, high false positive rates of the forecast results can be considered as the major problem; because wrong weather forecast results may make the civilian's day to day life style unstable and sometimes it can lead to huge financial losses in any country. Next section discusses the essence of ANN, GA and multi agent negotiations technology relevant to the weather forecasting domain.

3. Overview of Technologies Adapted

Through the literature survey, it was identified that the false positive rate of weather forecasting is a major problem. In order to produce a solution, essences of relevant technologies are discussed through this chapter with the justification of choice of a particular technology.

A. Artificial Neural Networks

The nature of the weather data available, they are incomplete, noisy and there is no theory to model the dataset and need generalization ability to predict weather results. According to the literature available [14], feed forward neural network model with a back propagation training algorithm is suitable to model weather forecasting models. When considering the domain of weather foresting, we can train a back propagation neural network by using historical data on weather conditions to predict future conditions. The trained neural network will be able to predict future weather conditions with the generalization ability.

B. Genetic Algorithm

When considering the problem of weather forecasting, we can set weather parameters such as Temperature, Relative humidity, Rainfall, Wind speed as genes to produce a chromosome in a genetic algorithm. Then by using fitness function and constraints on each gene in the chromosome with genetic algorithm we can produce high fit chromosomes as the forecasts for each parameter. Although the process takes more time to produce results it can generate adaptable results [15].

C. Multi Agent Negotiations for weather forecasting

In this work, planning to use the generic negotiation framework [16] to release the conflicts among ANN and GA based forecasting agents. There is a well defined protocol for agents to negotiate. One agent can be either ANN or GA based agent and calls out rainfall for next twenty four hours. The other party will respond to the proposal indicating whether it is acceptable or if not it will propose new rainfall forecast to the other party. This process goes on several times and both agents will populate the space of potential agreements (possible results). By using those potentially agreed forecast results; agents find the best result as the rainfall forecast. This best result finding process has been included in the negotiation protocol. Since this generic framework can be use differently to develop various negotiation applications, it has been used in an adaptable manner to solve the weather forecasting problem.

The next section discuss the main hypothesis of the project with explanations of how the selected technologies are used to solve the problem as inputs, process, outputs, users and features points of view.

4. An Approach to Using MAS Negotiations for Weather Forecasting

In order to produce a solution for the problem of false positive rate, rest of the work planning to propose an approach for rainfall forecasting using technologies mentioned in the previous section.

Our main hypothesis is the false positive rates problem in weather forecast reports which can be solve by using agent technology in the manner of multi agent negotiation among multiple perspectives.

The meteorological experts are considered as the users of the system. Proposed system uses minimum temperature, maximum temperature, relative humidity day time, relative humidity night time and rainfall for forecasting rainfall for next twenty four hours. The rainfall forecast report is generated separately from multiple perspectives such as ANN, GA associated agents. Most of the times these two agents will produce slightly different rainfall results for the same input and the ANN and GA forecasting models are not much accurate independently. In this situation the two agents identify their conflict. Then they start negotiation and deliberation process to share the findings of each agent for releasing the conflicts. In the negotiation both agents independently tries to reduce the disparity in results among each other and conclude the final results. This conclusion derives as the pair of results from ANN and GA where they have minimum difference.

The request, resource, message, ontology based multi agent negotiations architecture has been implemented through the work. The JADE framework used to develop the MAS. The main agents available in the system are GUI agent for interaction with the external users, ANN agent for forecast rainfall using artificial neural networks, and GA agent for forecast rainfall using genetic algorithm. In order for agents to operate, ontology is implemented in MATLAB. Additional knowledge for agents to operate developed as the Java rules available in a Java object. Finally, all the modules work together to form the multi agent negotiations based forecasting model.

Proposed system uses multiple technologies to forecast the rainfall. System has set of capabilities such as ability to work with incomplete and noisy data, produce results using limited data. It used both data and expert knowledge to produce results.

Since two different perspectives contribute for producing a single answer by using same environmental data set, it is expected to be a more rational rainfall forecast reports. Proposed request, resource, message space, ontology based multi agent system design presented in the next chapter.

5. Design of MAS Negotiations Based Weather Forecasting Model

This section elaborates the approach that was stated in previous chapter. The MAS based on the request, resource; message space and ontology architecture used to show the available agents in the system. Figure 1, Shows the interaction among each module.

A. Design of User Interface and User Interface Agent

This is the request Agent available in the system. It manages the user interface for users to interact with the system. At the starting of the GUI agent, the UI will be initialized and the ANN and GA based agents started. The GUI and GUI agent collectively support to gather inputs such as temperature, relative humidity and rainfall data in to the system. Then create the forecast request message and send the request to the ANN and GA based agents. It also supports displaying negotiation messages among resource agents with negotiated rainfall forecast related details.

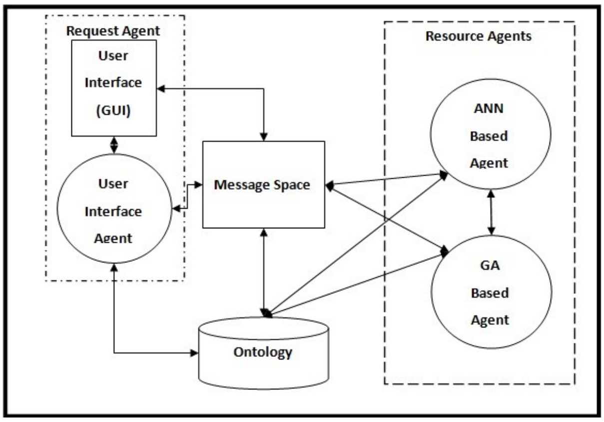

Figure 1: Multi agent negotiations architecture

B. Message Space

Message space is the place where all the agents write their important messages and read required details for the negotiation purpose. Information included in message space are input data such as temperature, humidity and rainfall and forecast results of each agent and other negotiation related messages of resource agents. Using message space all the agents are able to have awareness about current states of the counterpart agent. The agents in the system read the required messages from the message space when necessary.

C. ANN and GA Based Agents

The ANN and GA agents' uses current weather conditions in producing rainfall forecast for next twenty four hours. Once they receive a request for weather forecast from GUI agent, they query the ANN and GA models available in the ontology to produce initial rainfall forecast. Once the forecasts are available they update the message space by informing their current forecast results separately. Here after they pay attention to the counterpart agent's forecast results for same input conditions. Once the forecast of counterpart agent is available, it compares it with its own result. If there is a mismatch between results then that agent identifies the conflict and uses their negotiation knowledge available to start the negotiation process. The negotiation process continues until an agreement reaches or specified number of negotiation attempts reach. However ANN agent uses theoretical and heuristic foundation to conduct the negotiation while GA agent uses random search mechanism.

D. Ontology

The ontology maintains knowledge related to agents in the system. It helps to reduce the size of each agent by maintaining ANN and GA based forecasting models and negotiation related knowledge. ANN and GA based knowledge can be dynamically changed while the agents are in operation. Negotiation related knowledge is stored as rules in the ontology and this rule based knowledge does not change dynamically. All the agents have an ability to access their knowledge and dynamically make suitable changes on it.

1) Ontology Related to ANN

By default, a trained back propagation neural network is available in the ontology. ANN agent uses trained back propagation neural network model as the default model and it uses to generate forecast results at the beginning. Within the rest of negotiations, forecasting model is changed by the ANN agent for achieving more GA agent intensive forecast results. After the initial forecast, the dynamically changed forecasting model will be online trained by the ANN agent. In order to change the ANN forecasting models, the error goal, the learning rate and number of epochs should be changed by the ANN agent.

2) Ontology Related to GA

The GA based forecasting model is available in the ontology. This generates forecasting according to the forecast requests. When the negotiation goes on, this model generates more ANN agent intensive results to minimize the conflict among agents. For that, it continuously changes the crossover and mutation rate kind of GA options. When compared with the ANN forecasting model, the GA model is capable of producing not only the rainfall forecast but also temperature and relative humidity kind of other conditions for next twenty four hours simultaneously.

3) Rule Based Knowledge

The ANN agent maintains the negotiation knowledge as rules. These rules support to make suitable changes in ANN forecasting model to release conflicts among resource agents. These rules formatted as IF, THEN and ELSE manner.

Through this section design of the request, resource, message space, and ontology based multi agent negotiations architecture; their functions and interactions to produce rational forecast reports were discussed and Implementations of these modules will be discussed within the next section.

6. Implementation of MAS Negotiations Based Weather forecasting Model

The design has been implemented using software technologies such as Java technology, Java Agent Development Environment (JADE) and MATLAB tools.

A. Implementation of User Interface and GUI Agent

The UI was developed by using Java UI development components. This interface closely works with GUI Agent. This agent was implemented by extending the agent class available in JADE [17] libraries. Within the setup method of the GUI agent, UI will be called to start the interface and it initializes the agent platform controller. Once the platform controller is initialized, it then starts up the resource agents such as ANN and GA agents. The cyclic behavior class available in JADE [17] has been used to develop the mechanism which processes the incoming messages. This mechanism retrieves the messages and updates the GUI periodically.

B. Implementation of Message Space

Message space has a XML file which resides in the ontology. All the agents can access available nodes of XML file in the ontology by using document builder library in Java. Parameters such as Maximum Temperature (MaxT), Minimum Temperature (MinT), and Relative Humidity Day time (RHD), Relative Humidity Night time (RHN), Rainfall (RF), Rainfall forecast from ANN agent (RFTNN) and Rainfall forecast from GA agent (RFTGA) are stored in message space.

C. ANN and GA Based Agents

These agents were developed by extending the agent class available in JADE development library [17]. Each agent has two main types of behaviors. One for reading the receiving messages (ReadMessges) and the other for read the ontology (ReadOntology). The first behavior was developed by extending the CyclicBehaviour class available in JADE and latter was developed by extending the OneShotBehavior class available in JADE. The behaviors for reading ontology differ from agent to agent. Differences of resource agents depend on the functioning of these behaviors. The agents continuously read their own message queue to classify newly received messages from other agents. The current development contains two types of classifications namely REQUEST and INFORM. REQUESTs are for the received agent to do something such as generate a new forecast result using the forecasting models available in the ontology. INFORMs are for the received agent to get to know about something such as current results of others. According to the forecast results they decide to go for negotiations or conclude the final forecast.

The ANN agent and GA agent changes the properties of their own forecasting models to come up with the counterpart agent acceptable forecast result. In order to change the properties of the forecasting models in the ontology, the “eval” function which is available in matlabcontrolls library [18] was used within the agents. The ANN properties are the learning rate, error goal and number of epochs. GA agent changes its crossover and mutation options to come up with more agreeable forecasts.

D. Implementation of the Ontology

Implementation of the Ontology was based on MATLAB tools, XML and Java technologies. Separate MATLAB based forecasting modules are available for resource agents to use. A XML file maintains the message space related information inside the ontology. A java object holds the rule based knowledge for ANN agent to operate on negotiations.

1) Implementation of ANN Model

Implementations of ANN model was based on the Neural Network tool kit available in MATLAB [19]. Initially, a feed forward neural network with back propagation learning algorithm has used to forecast the rainfall. The implemented model contains five input neurons for input data such as maximum temperature, minimum temperature, relative humidity day time, relative humidity night time and rainfall. There are ten neurons available in the hidden layer and one neuron available in output layer which gives rainfall forecast as output. The log-sigmoid (logsig) transfer function was used as the activation function of the hidden layer. The linear transfer function (purelin) is used in the output layer. The resilient back propagation algorithm (trainrp) was used as the back propagation training algorithm [20]. Figure 2 shows the architecture of the ANN model.

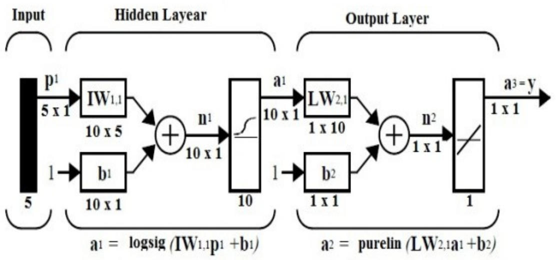

Figure 2: Architecture of the ANN model

2) Implementation of GA Model

GA tool kit available in MATLAB was used for develop the GA model. In order to support the negotiation process, the GA model is dynamically changed by the GA agent. The GA model contains a chromosome with six genes. The Figure 3 shows the chromosome structure.

MaxTF	MinTF	RHDF	RHNF	RFF	RFFDT

Figure 3: Structure of the chromosome in GA model.

The first five genes give the forecasts for the next twenty four hour conditions such as Maximum temperature forecast (MaxTF), Minimum temperature forecast (MinTF), Relative humidity day time forecast (RHDF), Relative humidity night time forecast (RHNF), Rainfall forecast (RFF) and the final gene gives the rainfall forecast of the day after tomorrow (RFFDT). The GA model consists of a mutable fitness function that has been designed heuristically and gene level constraints have been decided using available weather data set.

3) Implementation of Rule Based Knowledge

The ANN agent maintains a set of rules for negotiation purposes. In order to release the conflicts among ANN and GA agents, ANN agent uses available rules to change its own ontology (ANN forecasting model). These rules are encapsulated in to a java class. The rule outputs are based on the amount of difference between ANN and GA forecasting results. As shown in the examples following rules can be considered.

RULE 1:
IF number of epochs equals epochs has used in training THEN
Number of epochs equals epochs used in training plus Hundred
Update the current parameters
ENDIF

When the training process terminates due to number of epochs exceeded, it does not guarantee the proper convergence of the ANN model. In order to properly train the network it may need more epochs. So by increasing the number of epochs, we can expect a more converged network.

RULE 2:
If ANNForecast grater then GAForecast THEN
IF learning rate less than 0.08 THEN
Learning rate should increase by 0.01
ELSE
Learning rate equals 0.05
ENDIF

The RULE 2 is based on the back propagation algorithm [20]. When the ANN forecast is greater than GA forecast, in order to release the conflict, ANN forecast must decrease. According to the back propagation algorithm, increasing the learning rate causes to decrease the final answer of the used ANN model. This property has been used in this rule.

RULE 3:
If ANNForecast less then GAForecast THEN
IF learning rate greater than 0.01 THEN
Learning rate should decrease by 0.01
ELSE
Learning rate equals 0.05
ENDIF

The RULE 3 is based on the back propagation algorithm [20]. If the ANN forecast is less than GA forecast, in order to release the conflict among ANN and GA, ANN forecast must be increased. According to the back propagation algorithm, decreasing the learning rate causes to increasing the final answer of the used ANN model. This property has been used in this rule.

Next section uses implemented prototype for testing of project achievements against the project objectives.

7. Evaluation

This section is about testing of project achievements against the project objectives. It uses the implemented prototype MAS model as the major tool for demonstrating and testing the concept. The evaluation strategy has been designed to explain the level of achievements in objectives.

The weather forecasting models (ANN, GA) were trained using historical weather data sets based on Colombo city from 1st of October 2011 to 18th of November 2011. Twelve days of weather data from

19th November 2011 to 30th November 2011 has been allocated for testing purposes.

Rainfall forecast for 20th November, were generated on 19th November by issuing weather conditions of the 19th November. Then the database was updated (data used to train the ANN model) on 20th November by using 19th November measured conditions and 20th November measured rainfall. Now database was then ready to forecast rainfall for 21st November by using the 20th November weather measurements. This process was conducted until the rainfall forecast of 30th November. Day by day the setup was updated to have up to date data for training the ANN model.

In order to forecast the amount of rainfall for next twenty four hours, number of negotiation attempts has been constrained to ten. In this experiment it selects the best forecast result on a daily basis. The negotiated rainfall forecast results of ANN and GA agents have been plotted for showing the negotiation process among ANN and GA agents and behavior of disparity between each other. The reports on collected results presented in Figure 4 and selected best forecast results for each day available in the Figure 5.

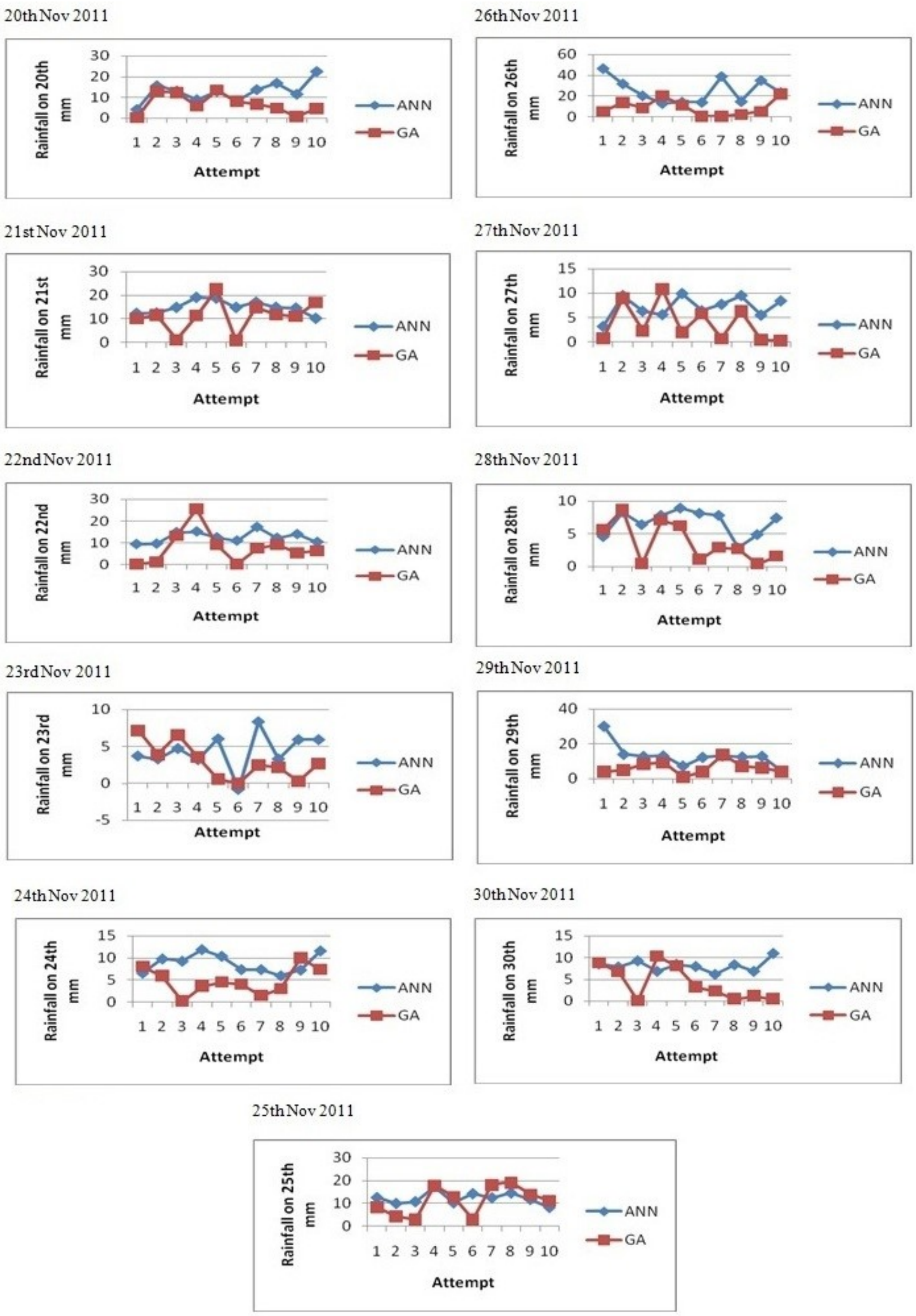

Figure 4: Experimental results with ten negotiation attempts

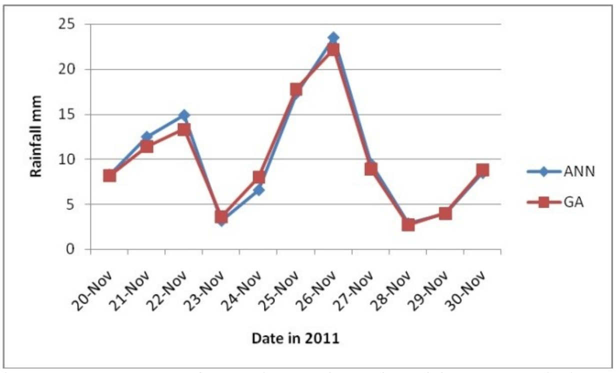

Figure 5: Summary of experimental results with ten negotiation attempts

According to Figure 5 the mean square error of results between ANN and GA solutions was 0.76. The results obtained from the evaluation will be interpreted to analyze the strength of the proposed solution in the next section.

8. Conclusion

This section focuses on the interpretation of the results that were taken from the evaluation. Finally, the section discusses about potential further work which will be worth doing as the continuation of the project.

According to the experimental results that were mentioned in the previous chapter, it shows the gradual reduction of disparity of forecast results between ANN and GA agents. Further it shows that when the number of negotiation attempts increases, the disparity between two agents reduces. The experiments conclude when solutions by ANN agent and GA agent shows a disparity at the beginning, but reach to commonly agreeable solution through the negotiation in the multi agent solution with around 65% of success.

Through this work a MAS negotiations based multiple perspective approach to real world problem solving was discussed. The weather forecasting domain was used as the application area to show the proposed concept. There are two soft computing approaches (ANN and GA). They have been employed to produce results within the agent negotiation environment. By increasing the number of forecasting technologies, it can be guaranteed to have more rational and quality results. The statistical weather forecasting approach called Analog weather forecasting approach and a decision tree based forecasting model can be proposed to use in separate agents as further work. By enabling MAS negotiation among more approaches or technologies, it can guarantee the accurate results beyond the capabilities of each individual agent.

Acknowledgement

Authors wish to acknowledge the Department of Meteorology Sri Lanka for providing historical weather data for the research and all who supported in any respect during the completion of the work.

References

[1] Weather.org. (undated). World Weather Forecasts [Online]. Viewed 2011 Oct 25. Available: http://weather.org/

[2] Tiruvenkadam Santhanam and A.C. Subhajini (2011), “An Efficient Weather Forecasting System using Radial Basis Function Neural Network,” Journal of Computer Science 7 (7), pp 962-966.

[3] N. Q. Hung, M. S. Babel, S. Weesakul, and N. K. Tripathi (2009), “An artificial neural network model for rainfall forecasting in Bangkok, Thailand”, Hydrol. Earth Syst. Sci., 13, pp 1413–1425.

[4] Imran Maqsood, Muhammad Riaz ,Khan Ajith Abraham (2004), “An ensemble of neural networks for weather forecasting”, Neural Comput & Applic.

[5] LL. LAI (August 2004), “Intelligent Weather Forecast”, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai.

[6] Kwok Yuen Keith, “Hybridized Genetic Algorithm and K-Nearest Neighbor for rain fall prediction”, B.C.Sc.(Hons) dissertation, Department of Computer Science, RMIT, March 1994.

[7] C. M. Kishtawal, Sujit Basu, Falguni Patadia, and P. K. Thapliyal (2003), “Forecasting summer rainfall over India using genetic algorithm”, Meteorology and Oceanography Group, Space Applications Centre, Ahmedabad, India.

[8] Bjarne K. Hansen and Denis Riordan (2001), “Weather Prediction Using Case-Based Reasoning and Fuzzy Set Theory “, Faculty of Computer Science, Dalhousie University, Canada.

[9] S. Santhosh Baboo and I.Kadar Shereef (October 2010), “An Efficient Weather Forecasting System using Artificial Neural Network”, International Journal of Environmental Science and Development, 1(4), pp 128-133.

[10] James N. K. Liu and Raymond S. T. Lee (1999),“Rainfall Forecasting from Multiple Point Sources Using Neural Networks”,IEEE,.

[11] Juan Peralta, German Gutierrez, Araceli Sanchis, (2010), “Time series forecasting by evolving artificial neural networks using genetic algorithms and estimation of distribution algorithms” ,IEEE World Congress on Computational Intelligence.

[12] Raymong Lee and James Liu (August 2004), “iJADE WeatherMAN”: A Weather Forecasting System Using Intelligent Multiagent-Based Fuzzy Neuro Network”, IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 34(3), pp 96-102.

[13] Luis, Cesar and Oscar, “Forecasting of a Chaotic Time Series with Artificial Intelligence Techniques”, The

Online Journal on Electronics and Electrical Engineering (OJEEE), 2(1).

[14] RC Chakraborty, “Back Propagation Networks”,(2010),”Soft Computing Course Lectures.

[15] Lary yaeger , “Intro to Genetic Algorithms”, Indiana University.

[16] N. R. Jennings, P. Fartin, A,R. Lomuscio, S. Parson, M. Wooldridge, C. Sierra,(2001) “Automated Negotiation: Prospects, Methods and Challenges”, Group Decision and Negotiation 10,pp 199 – 215.

[17] Fabio Bellisemind et al. “JADE Programmer’s Guide”, University of Parma, 08 April 2008.

[18] University of Verginiya (undated). “Calling Matlab from Java “[Online]. Available: http://www.cs.virginia.edu/~whitehouse/matlab/Java Matlab.html

[19] MathWorks. (Undated). “Neural Network Tool kit” [Online]. Viewed 2011 Oct 25. Available: http://www.mathworks.in/help/toolbox/nnet/index.htm l

[20] Howard Demuth,Mark Beale, “Backpropagation,” in Neural Network Toolbox, 4th ed. The MathWorks, Inc, 2002, ch. 5, sec. 4, pp. 132–152.

Framework for Discovery of Data Models Using Genetic Programming

W.J.L.N Wijayaweera[1], A.S. Karunananda[2]
Faculty of Information Technology, University of Moratuwa
Sri Lanka
[1]njaymax@gmail.com, [2]asoka@itfac.mrt.ac.lk

Abstract—**The field of Genetic Programming in Artificial Intelligence strives to get computers to solve a problem without explicitly coding a solution by a programmer. Genetic Programming is a relatively new technology, which comes under automatic programming. After the initial work by John R. Koza in genetic programming, much research work have been done to discover data models in various datasets. These work have been rather domain specific and little attention has been given to develop generic framework for modeling and experimenting with genetic programming solutions for real world problems. This paper discusses a project to develop a visual environment, named as GPVLab, to design and experiment with genetic programming solutions for real world problems. GPVLab has successfully discovered data models for various data sets and according to the main evaluation it is evident that GPVLab can generate solutions which provide better results in 56% of the time. It is concluded that GPVLab can be used to model genetic programming application very conveniently. GPVLab can be used not only for discovering data models but also doing various experiments in genetic programming.**

Keywords: **Genetic Programming, Artificial Intelligence,Automatic programming.**

1. Introduction

Nature develops everything by the means of natural selection and constant evolution. This is the inspiration behind genetic algorithms that mimics the process of natural evolution [1]. In last few decades of twentieth century the field of genetic programming (GP) has been developed, mostly by John R. Koza, as an extension of genetic algorithms (GA) [2]. GP has repeatedly proved to be a much better approach in solving most symbolic regression problems [3]. Also this field helps to make programs which may automatically generate solutions for a particular problem. In Artificial Intelligence this capability is known as automatic programming which attempts to get a computer to solve a problem without explicitly coding a solution by a programmer. GP is considered as a type of automatic programming which inspired by the Darwinian evolution [4]. It progressively breeds a population of computer programs over a series of generations [5].

A project has been launched to develop a visual environment to design and experiment with genetic programming solutions for real world problems. It is named as GPVLab to mean its ability to facilitate the discovery of data models for real world problems through a wide range of experiments in a visual environment. GPVLab addresses the difficulty of discovering a data model in any numerical dataset and automatically generates an evaluable expression to mimic the discovered data model. It is able to discover data models from any numerical dataset, even with noisy data. It also facilitates to conduct genetic programming experiments. Nevertheless GPVLab is user friendly and it can be used by any person even without the genetic programming knowledge.

GPVLab takes any numerical dataset as the input. Users can initiate the main process by feeding data and selecting the reference column. Then the system runs the genetic programming process by generating populations of expressions and evaluating them to find their fitness. Finally the system determines the best fit model for the dataset. Upon completion of the discovering process, GPVLab immediately allow users to evaluate the model by providing required parameters. It has the facility to save resultant models and access and evaluate them via model library as required. GPVLab has been developed in C# language using Microsoft Visual Studio 2010 [6] and Microsoft .NET Framework 4.0 [7]. AForge.NET 2.2.3 Framework [8] is used as the Genetic Programming toolkit. Furthermore GPVLab model library has been developed using a SQL Server Compact 3.5 [9] database.

In the next section, related work in discovering data models will be discussed. Section 3 discusses the technology behind this project. Section 4 presents genetic programming approach to discover data models. Furthermore section 5 discusses the design of the GPVLab and section 6 discusses the implementation of GPVLab. Section 7 presents the details about evaluation of GPVLab. Finally section 8 present the conclusion and further work related to this project.

2. Related Work in Discovering Data Models

In the field of Artificial Intelligence (AI), machine learning approaches are popular for problems related to regression analysis. Generally most of the researchers use Artificial Neural Network (ANN) and Genetic Algorithm (GA) based solutions for regression analysis. In both occasions an insight of the data is necessary even before proceeding to create a good design. ANN does not promise to converge into a good structure if layers and nodes are not carefully designed. It is hard to determine the optimal value for number of hidden layers needed for specific problem. However, it is possible that if ANN is used, in order to get good results, one may have to redesign the whole system for different datasets rather than training the same system with new data. This may not provide the best answer for typical real world datasets, which we have gathered through various researches and surveys. Nevertheless there are few attempts been made to develop ANN based systems which can eliminate most of these limitations.

D. F. Specht proposes a new memory-based artificial neural network (ANN) that converges to the underlying (linear or nonlinear) regression surface [10]. The new network is called a general regression neural network (GRNN) which is a one-pass learning algorithm with a highly parallel structure. He argues that the algorithmic form of this solution can be used for any regression problem. The resultant system can be used for prediction, modeling, mapping, and interpolating or as a controller [10]. However; this sort of systems are not suitable for problems where we do not have any prior knowledge on the structure of the resultant model.

GA is not always being a good solution for regression problems because it needs a good fitness function prior to development. Nevertheless several attempts have been made to improve GA based regression analysis systems. S. Tomioka and et al. proposed a new method, adaptive domain method (ADM), to change the domain dynamically using several Genetic Algorithms with short lifetimes [11]. They have applied this to overcome a disadvantage of applying several genetic algorithms (GAs) for nonlinear least square (LS) regressions problems. Furthermore they have demonstrated the improvements in terms of both the convergence and the reliability by ADM. Another advantage of this proposed method is that it does not require any specialized knowledge about GAs or their tuning [11]. Nevertheless these kinds of systems are rather difficult to implement and these are not capable of generating evaluable expressions corresponding to the resulting model.

It is noteworthy that there are some other non-AI approaches to develop such systems as well. Surprisingly some of these systems outperform their AI related counterparts. S. Crino and D. E. Brown presented a procedure for approximating the global optimum in structural design by combining multivariate adaptive regression splines (MARS) with a response surface methodology (RSM) [12]. They argue that even though there are many global optimization routines, few are appropriate for expensive deterministic simulation models such as those used in structural design optimization. According to them the MARS/RSM procedure is a suitable replacement for inefficient techniques such as genetic algorithms (GAs) and simulated annealing (SA). MARS/RSM efficiently and accurately converges to a feasible solution by applying variable screening, domain reduction, and an efficient experimental design [12]. Nevertheless these systems require prior knowledge on the structure of the resultant model.

It is clear that there is a need of a good technology which can automatically identify relationships among attributes in a given dataset. This enables computer systems to discover new models without any intervention of humans. This sort of technology is really important to analyze data in non technical fields like agriculture. That is because the researchers who are working on those areas do not need to have a technical knowledge which is not relevant to their field of research; therefore they can concentrate more on their research rather than tackling technical issues.

In nature everything develops over time through evolution. We can see that nature is using a development approach which grows from the beginning. Nevertheless, we use a compositional approach rather than using developmental procedures [4]. This is rather unnatural, and we still do not know how to grow a complex device like a computer. In the early 20th century, Darwinian theories of evolution formed the idea of development by the means of natural selection and constant evolution. This is the inspiration behind the introduction of genetic algorithms, by John H. Holland, which mimics the process of natural evolution [1]. Until then we were using a compositional approach in every aspects of programming [4]. Nevertheless genetic algorithms do not help to grow a complex structure like a computer program from scratch, which is one of the central goals of computer science, called automatic programming. It is also known as automatic induction of machine code. If the performance of that automatically generated program equals or exceeds the human level, we say it is useful [4]. Nevertheless current methods are not mature enough to develop complete automatic programming systems. According to Arthur Samuel's definition the goal of automatic programming is to "Tell the computer what to do, not how to do it" [13].

In Arthur Samuel's work he mentions the possibility of developing computer programs which will learn and perform better than the person who wrote the program. His work shows that a computer is learning how to play checkers and eventually it performs better than the person who has developed it [13]. If we develop a program which can behave like human being or an animal is doing that particular task, then we can say that process involves learning. Therefore we have to build a program which can learn from available data and evolve. This will eventually eliminate the need of detailed programming effort.

R. Balzer elaborated a simple notion of an automatic compiler into a paradigm for automated software development [14]. He has identified two fundamental flaws in the current software paradigm as the lack of technology for managing the knowledge intensive activities and the maintenance is always performed on source code. He has showed that his proposed paradigm, even without full automation, eliminated those flaws in the current paradigm. It is rather difficult to develop such systems using conventional technologies. There isa need of a new technology which can be easily implemented and capable of producing novel programs to solve significant problems.

Most of the existing paradigms by 1980's involve specialized structures which are nothing like computer programs [2]. When getting computers to solve problems without being explicitly programmed, it should not be required to specify any attributes related to the solution in advance. Instead, these attributes of the solution should emerge during the problem-solving process. Therefore researchers started to follow the principles of nature instead of the approaches that considered correctness, consistency, justifiability, certainty, orderliness, parsimony, and decisiveness of the solution [2].

Generally if we need to get computers to solve problems without being explicitly programmed, we have to automatically generate programs using segments of basic programs [2]. The first inspiration for automatic generation of simple sequential programs has given by Nichael L. Cramer in 1985 [15]. In his work he stated that even though genetic algorithms have been proved to be useful in many areas, most of these systems focus on adjusting fixed set of parameters in order to optimize the performance of a given algorithm. He clearly explains the possibility of using crossover, selection, mutation, and tree representations to generate programs using genetic algorithms. He shows the possibility of developing simple sequential computer functions, starting with low-level computational primitives, by automatically generating a well-defined two-input, single-output multiplication function [15].

After Cramer's initial work [15], John R. Koza greatly expanded that work to form the modern programming paradigm called genetic programming (GP) [2]. GP is also based on the Darwinian principle of reproduction and survival of the fittest [5]. Even though GP is an extension of Genetic Algorithms, in GP, we evolve a population of computer programs. This automatic problem solving nature of GP does not require the user to know or specify the form or structure of the solution in advance [16]. Hence, GP became the first programming paradigm which is heading towards automatic programming. Since its introduction, GP has been very successful at evolving novel and unexpected ways of solving problems [16].

GP is a form of automatic programming which was inspired by the Darwinian evolution [4]. GP maintains the genetic algorithms' overall algorithm. It progressively breeds a population of computer programs over a series of generations [5]. The dynamic variability of the automatically generating computer programs is an important feature of GP [2]. GP is not a fully automatic programming approach which can generate any computer program [4]. For example it cannot generate a word processer. We use GP as a problem solver, a function approximater, and an effective tool for writing functions to solve specific tasks. It does not replace programmers but it helps them. Programmers should specify the fitness function and identify the problem to which GP should be applied. We should look at GP as another tool for programmers [4]. Over time GP proved its capability of producing human-competitive results [4]. It has produced human-competitive results in areas such as control, design, classification, pattern recognition, game playing, and algorithm design. This became possible with use of the LISP programming language which supports the processes in genetic programming. The symbolic expressions (S-expressions) of the LISP are an especially convenient way to create and manipulate the compositions of functions and terminals (input to the computer program to be discovered) [17].

GP produces really good results in the areas where the interrelationships among the relevant variables are poorly understood and areas where there is a large amount of data in computer readable form that requires examination, classification, and integration [2]. GP is also good at unveiling unexpected relationships among variables and producing analytical solutions where conventional mathematical analysis cannot be used [16]. Applications of GP range from small subroutines to real-time problems and offline applications such as time-series analysis or data-mining tasks [16]. GP techniques have been used for the evolution of structures other than computer programs, such as neural networks and analog electrical circuits [5]. GP has produced human competitive results in areas such as control, design, classification, pattern recognition, game playing, and algorithm design [4]. GP is beginning to find interesting new programs that humans had not previously discovered and it is becoming more and more popular in solving symbolic regression problems.

D. A. Augusto and H. J. C. Barbosa presented implementation for solving symbolic regression problems using genetic programming (GP) [3]. Since the standard implementations of GP in compiled languages are not much efficient, they have presented a new approach using Read's linear code. They have presented observational data to prove that this approach lead to more simplicity and better performance compared to traditional GP implementations.

H. Tuan-Hao and et al. have presented some experimental results using Incremental Evaluation with Tree Adjoining Grammar Guided Genetic Programming (DEVTAG) [18]. They have used symbolic regression problems such as Fourier series problem (sawtooth problem) and a benchmark polynomial fitting problem in genetic programming. In their study they have presented comparison results to show that the results of their approach outperformed both standard Genetic Programming (GP) and the original Tree Adjoining Grammar Guided Genetic Programming (TAG3P). However they have not directed their study towards a general system which can solve any symbolic regression problem.

A similar work has been conducted by G. Dworman and et al. to discover high-quality negotiation policies using genetic programming [19]. They have presented series of successful experimental results. Nevertheless this study has not directed towards developing a system which can solve symbolic regression problems by working with any input dataset.

We have encountered two similar products related to GP, namely GPdotNet [20] and GPLab [21]. GPdotNet is free software which was developed using Microsoft .NET technologies. This software is capable of handling large datasets and this supports multi core parallel processing as well. Nevertheless this software is not user friendly and it lacks the noise reduction capabilities. GPLab is a relatively old toolbox for the MATLAB [22] or Octave [23]. GPLab is an advanced tool for researchers who are doing advanced experiments using genetic programming. Since this is not a standalone application but a toolbox for MATLAB, the users should have the knowledge on MATLAB or Octave as well. This toolbox also does not provide noise reduction as a feature.

It is evident that enough work has been done to discover models from datasets using genetic programming. However, there is little or no work has been done towards developing a general solution which can discover data models even from noisy numerical dataset. Such a user friendly general program is an ideal solution for non-technical domains. Considering all these facts we can see that that there's huge potential that we can use GP to unveil the relationship among attributes in any numerical dataset. Therefore if we can develop a software which can take datasets as inputs and produces computer programs to represent the data model in the input dataset, it makes it possible to discover previously unknown data models as well as readily predict the outcomes of future events using the automatically generated computer program.

3. Technology Adapted

Genetic programming (GP) is a domain independent problem solving approach in which computer programs are evolved to solve, or approximately solve, problems. In each generation, GP stochastically transforms populations of programs into new, optimistically better, populations of programs. GP is a random process, and it can never guarantee results. Nevertheless GP's randomness can lead it to go beyond the capabilities of existing paradigms and solve problems in unexpected ways.

GP finds out the performance of a program by running it, and then comparing its behavior to some ideal situation. That means GP can search through the set of possible computer programs to find an individual computer program which has the highest fitness in solving, or approximately solving, the problem at hand. The fitness measure is the most important fact which produces the needed program structure. The final computer program structure that emerges from GP process is a result of this fitness.

In GP, user must perform five major preparatory steps before applying GP to a problem. First step is to determining the set of terminals. The terminals are the inputs to the resultant computer program. Secondly the user should determine the set of primitive functions. These are the set of functions that are to be used to generate the mathematical expression that attempts to fit the given sample of data. Third step is to determine the fitness measure which decides how well each individual computer program, in the population, performs in the particular problem environment. Each individual computer program in the population is executed and then evaluated using this fitness measure. The fourth step is to determine the parameters which control the execution of the GP process. The primary parameters for controlling a run of GP are the population size and the maximum number of generations to be run. The fifth and last step is to determine the method for designating a result and the termination criteria. One frequently used method of result designation is to designate the best individual obtained in any generation of the population as the result for that generation. Termination criteria can be taken as the event of exceeding the maximum number of generations.

In main GP process there are three major steps. First step is to generate an initial population of random compositions of the functions and terminals of the problem. Secondly iteratively perform two sub steps until the termination criterion has been satisfied.

1) Execute each program in the population and assign it a fitness value.

2) Create a new population of computer programs by applying genetic operations. The genetic operations are applied on selected computer programs from the population. Selection process is based on fitness of the each program.

There are three main genetic operations. First operation is the Darwinian reproduction which is also known as cloning. This operation reproduces an existing program by copying it to the new population. Second operation is crossover which creates two new computer programs from two existing programs by genetically recombining randomly chosen parts of two existing programs. Mutation, which is the third operation, creates one new computer program from an existing program by mutating a randomly chosen part of the program. The third and final step is to output the best-so-far individual program as the result. This result may be a solution, or approximately a solution, to the problem.

4. A Genetic Programming Based Approach to Discover Data Models

It is apparent that GP allows us to discover data models and automatically generate evaluable expressions to evaluate those models using any given dataset. It is possible to use any numerical data set as an input to this software system. This system should allow the users to load data using files in a convenient file format such as comma separated value (.csv). It also should facilitate the users to manually enter required data. Furthermore the user should select the reference column before starting the discovering process. If user has not selected any specific column, the system should take the last column on the dataset as the reference column. Furthermore advanced users may change parameters related to the genetic programming process. These parameters may include control parameters related to reduce noise in data as well.

When user initiates the process by entering the data set and selecting the reference column, the system will run the genetic programming process by renaming the attribute names. The system will randomly generate populations of expressions and evaluate them to find their fitness. After going through generations after generations, the system will determine the best fit model for the dataset. If the maximum number of generations exceeded the system will stop the process and output the best so far model. Once the main process is completed, the system should output the discovered model as an expression. This expression can be evaluated as needed by providing values for required parameters. Furthermore users should have the facility to save the expression. Once saved, this expression can be evaluated at any time by entering values for required parameters. Furthermore the user can export the data model into a more portable form which can be evaluated the model independently.

User for this system can be any person who needs to discover a model out of a collected numerical data set. Advanced users who have knowledge about Genetic Algorithms of Genetic Programming can try changing advanced parameters related to the genetic programming and noise reduction for better results. Nevertheless the default settings will work for almost all the problems. Knowledge about Genetic Programming is not a necessity.

This software should have the features such as, the ability to create a dataset through the system and directly use an existing .csv file to load a dataset. This software also provides advanced Genetic Programming settings for advanced users. It has a feature to save output expressions and evaluate saved expressions at any time by providing values for required parameters. This system allows users to export expressions as well. This feature should allow users to export saved models to independent and portable executables.

5. Design of the Genetic Programming Based Software Solution

This software solution has been named as GPVLab to mean its ability to facilitate the discovery of data models for real world problems through a wide range of experiments. In order to build a user friendly system, user should have a proper way to interact with the system and enter data into system. Once the data is available the main module of GPVLab should have the facility to perform genetic programming process using available data. Once the process is completed GPVLab should output the resultant expression. GPVLab should have the facility to evaluate the resultant expression by entering values for required parameters. Also GPVLab should facilitate to save and export resultant expressions for evaluate them as needed. Fig. 1 shows a detailed design diagram of this software solution.

In the initialization stage system should gather all the relevant information to perform the model discovery process. In this stage GPVLab allow the user to either create a dataset from the beginning or load data from an existing file. The file should be in a convenient format for both user and the system, such as .csv. Furthermore GPVLab should allow users to specify the reference column. If the reference column is not specified the system should automatically select the last column as the reference column. GPVLab should facilitate the users to specify values for most important parameters which are directly involved in Genetic Programming process. However it should specify default values for all these parameters in an optimal manner to make the system usable for average users.

Once the required data and other additional parameters have been set, GPVLab should begin the genetic programming process by generating an initial population. This population consists of randomly generated evaluable expressions. Each evaluable expression is a valid mathematical expression which contains operators and operands. Once the user starts the main process, column names of the input dataset (except the reference column name) will be taken as the terminals. Set of constants along with these column names will create the set of possible operands. Then the system will take arbitrary number of operands (from the operand set) and randomly selected operators to make valid expressions. After a validation process, this newly created evaluable expression will be put into the population. Once the initial population is generated, it should perform the selection of the best fit expression by executing each of them and evaluating the fitness measure of those expressions. Then it should populate the next generation by performing genetic operations (cloning, crossover and mutation) over selected expressions. This process will iteratively be continued until the termination criterion has been satisfied. Finally GPVLab should give the exact expression or the best so far expression as the solution.

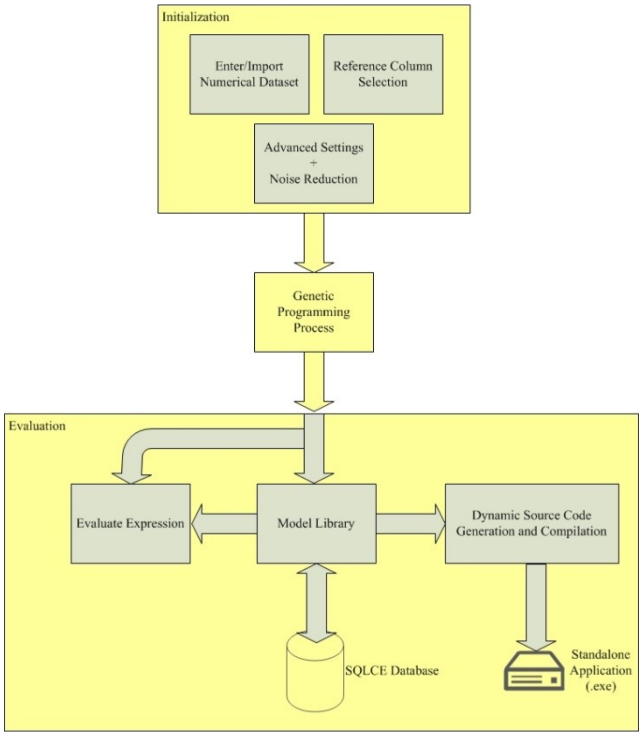

Figure1: Detailed design diagram of this software solution

As soon as the discovery process is completed, users should have the facility to evaluate this expression through the system. GPVLab should also facilitate the users to save the expressions for future reference. Inbuilt facility should be provided to browse and evaluate saved expressions. At the time of execution of the resultant expression, system should prompt the users to enter values for required parameters. Once the required parameters have been assigned with user input values, the system should execute the expression and the result of the evaluated expression should be present to the user. Furthermore GPVLab should have the facility to export saved data model. This facility can be provided by dynamically generating a code and compiling it to form a standalone executable. This feature is important when user need a portable simple application that does not depend on the main software system. Therefore user should have the ability to generate portable standalone applications from saved data model expressions.

6. Implementation of GPVLab

GPVLab has been developed using C# language with .NET framework 4.0. The AForge.NET library has been used as the main framework for genetic programming. GPVLab also extends some classes of AForge.NETframework to accommodate data with arbitrary number of attributes and to remove noise in data. GPVLab has been implemented as a MultipleDocument Interface (MDI) application. Inside the GPVLab main window, users can open multiple instances of explorer windows. Each explorer window has the facility to independently conduct experiments using numerical datasets. This feature is very important in performing parallel experiments. Fig. 2 shows a screenshot of an explorer window. GPVLab's Explorer window takes any numerical dataset as the input. There are two possible ways to feed datasets into the system. Inside the explorer window, either user can enter a required data using the inbuilt facility or import a dataset from existing comma separated value (*.csv) file. These options are available inside the "Data Input Methods" section of the explorer window. If user selects to manually enter data, the system prompts the user to enter the number of columns for the dataset. Then GPVLab will generate a dataset with specified number of columns with automatically generated column names. Once the dataset is entered or imported into GPVLab, user should select the reference column from the given drop down list. By default GPVLab automatically selects the last column as the reference column.

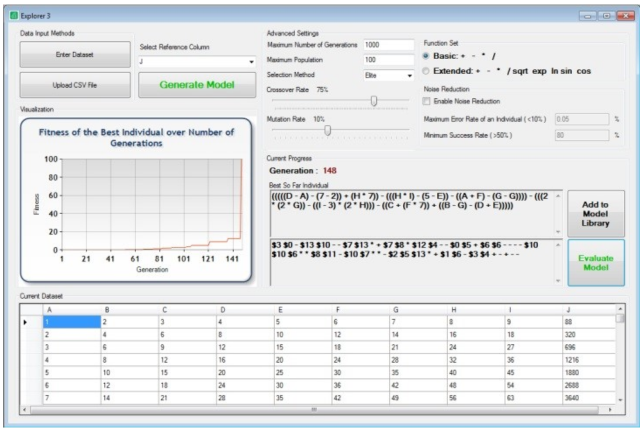

Figure2: A screenshot of an explorer window

Furthermore an advanced settings section is provided for advanced users, where control parameters of the genetic programming process can be adjusted. This section includes settings related to the noise reduction feature as well. Advance settings section contains six major parameter settings which directly involves in the genetic programming process. First the user can define the maximum number of generations to be run. This is one of the termination criteria of the geneticprogramming process. Next, users can adjust maximum number of population. Then user can choose the selection method. It contains three selection methods, namely "Elite", "Rank" and "Roulette Wheel". The system also allows the user to select the mathematical function set to be used in the genetic programming process. The system has an option to select one out of two function sets for this purpose, namely basic and extended. Basic function set contains operators such as addition, substations, multiplication and division. Extended faction set has the ability to generate models consisting of sqrt (Square root), sin, cos, ln and exp (exponential) in addition to the basic operators. Finally the advanced settings section facilitates the user to adjust the crossover and mutation rates.

Furthermore if the data set contains noisy data, users have the facility to enable inbuilt noise reduction feature. Once this is enabled the system allows users to change two additional parameters related to the noise reduction. Those are the maximum error rate and the minimum success rate. If the noise reduction is enabled,those two parameters also affect the genetic programming process. Once the initialization phase is over, the user can start the model discovery process by clicking on the "Generate Model" button. This starts the genetic programming process. Genetic programming process generates the initial population by randomly creating expressions consist of operators from the selected function set, terminals and constant set. Terminals will be the column names other than the reference column. The constant set contains values such as 1, 2, 3, 5 and 7. Fig. 3 gives some important code segments involved in main genetic programming process. Once this process is started GPVLab displays the current generation and the real time visualization of the change in fitness of the best individual for each generation.

Once the genetic programming process starts; it will continuously breed evaluable expressions. Each of these individuals is evaluated for its fitness. First of all, an individual is validated to see whether it consists of all the required terminals. Once an individual passes the validation check, that particular individual is evaluated for the entire dataset to get the total error. This is performed by evaluating the individual for each record in the dataset. Error of an individual is determined by calculating the absolute difference between the result it gives for a particular record and the corresponding reference column value. If the noise reduction is enabled the error is calculated in a different way. If the error of the individual is in the acceptable range, which is defined as maximum error of an individual, then this individual is counted as a success for that particular record in the dataset. This is performed for the entire dataset and calculated for the number of successful encounters within the dataset. Furthermore if the noise reduction is enabled, before returning the fitness value, a method checks the eligibility of that particular individual. If the individual is eligible, which means it achieved the minimum success rate, and then this method adds this expression to the list of successful individuals.

The fitness function uses in the genetic programming process is "100 / (1 - sum of absolute errors)". If the error is zero then this makes the fitness value of 100 for that particular expression, which means a perfect solution. If the system has found a perfect solution, the genetic programming process will immediately end and resultant expression is presented to the user. If a perfect solution has not been discovered within the given number of generations , then the GPVLab outputs the best so far expression. Either way the system outputs the best so far evaluable expression in reverse polish notation (RPN). If the resultant expression contains only basic operators, GPVLab automatically converts the resultant expression in RPN notation into a human readable expression in infix notation. Furthermore if the noise reduction feature is enabled, the expression with the lowest operator count, out of eligible expressions, will be considered as the best solution. Nevertheless GPVLab may stop the process and output the solution if the genetic programming process founds a 100% successful solution for the particular dataset.

Upon completion of the discovering process, users can immediately evaluate the model by providing required parameters. Also GPVLab has the facility to save resultant models and access and evaluate them via model library as required. Model library is developed using SQL compact edition database, which does not require SQL Server instance to operate. Hence, GPVLab is highly independent and can be installed or run on any computer with just .NET Framework 4.0 installed. Fig. 4 shows a screenshot of the model library. Furthermore the model library also provides a facility to export saved expressions as standalone executables (.exe). This has been achieved using the Microsoft CodeDom technology. This mechanism generates a highly portable executable, which can be executed as needed on any computer with just .NET Framework 4.0 installed.

7. Evaluation of GPVLab

The main experiment has been done using a real world noisy dataset. The data was taken from the World Bank Data Catalog [24]. This repository contains data about each country in the world. For this experiment author has taken the data sheet under the "World Development Indicators" section in Sri Lanka. This data sheet contains various indicators about the country's development. For the purpose of this experiment author has taken eight important data columns. These are "Total reserves (includes gold, current US$)" (TR), "Inflation from consumer prices (annual %)" (I.CP), "General government final consumption expenditure (% of GDP)" (GGFCE), "Exports of goods and services (current US$)" (EOG.S), "Gross domestic product (current US$)" (GDP), "Official exchange rate (LCU per US$, period average)" (OER), "Life expectancy at birth" (LEABT) and "Total population" (PT). Apparently there is no relationship between these data columns. Out of all the data author has taken the fifty records from year 1960 to 2010 for this experiment.

The purpose of this experiment is to generate an expression to find the GDP. Further this experiment compares the result of GPVLab with the result of a conventional data modeling mechanism. For this purpose author has selected the Linear Regression functionality in WEKA (Waikato Environment for Knowledge Analysis) [25]. First author imported the dataset into GPVLab. Then the author has selected the "Gross domestic product (current US$)" (GDP) as the reference column. After selecting the 'GDP' as the reference column, without changing default settings, the genetic programming process has been initiated by clicking on the "Generate Model" button.

The main genetic programming method needs the input dataset as a multidimensional double array. Therefore the following code segment is used to convert the data in the data grid into a multidimensional double array.

```
double[,] InputDataset =
CommonUtilities.DataGridViewToDoubleArray(dgvInputData);
```

Following code segment initializes the fitness function.

```
ExtendedSymbolicRegressionFitness Fitness = new
ExtendedSymbolicRegressionFitness(InputDataSet,
InputConstants);
```

Following code segment is used to initialize the population.

```
IGPGene Gene = (IGPGene)new
SimpleGeneFunction(NumberOfConstants + ArgumentsCount);
if (FunctionSet == 1) {
  Gene = (IGPGene)new
ExtendedGeneFunction(NumberOfConstants +
ArgumentsCount);}
```

Creating the Chromosome

```
IChromosome GPChromosome = (IChromosome)new
GPTreeChromosome(Gene);
```

Initializing the selection method

```
ISelectionMethod GPSelectionMethod = (ISelectionMethod)new
EliteSelection();
if (SelectionMethod == 1){
  GPSelectionMethod = (ISelectionMethod) new RankSelection(
); }
else if (SelectionMethod == 2){
  GPSelectionMethod = (ISelectionMethod) new
RouletteWheelSelection( );}
```

Initializing the Population

```
Population GPPopulation = new Population(MaxPopulation,
GPChromosome, Fitness, GPSelectionMethod);
```

Setting Crossover Rate

```
GPPopulation.CrossoverRate = CrossoverRate;
```

Setting Mutation Rate

```
GPPopulation.MutationRate = MutationRate;
```

Once the population is initialized following code segment is used for run the genetic programming process for a single generation.

```
GPPopulation.RunEpoch();
```

Finally we can access the best so far individual using following code segment.

```
BestSoFarIndividual =
GPPopulation.BestChromosome.ToString();
```

Figure3: Important code segments of the main method related to the genetic programming process.

At the beginning GPVLab couldn't find a good solution with normal settings. After several experiments GPVLab has found a solution within 132 generations. GPVLab has achieved this after setting maximum number of generations to 10000, maximum population to 500 and maximum error rate of an individual to 25. Please note that extended function set was selected and noise reduction was enabled. All other settings were set to their respective default values. GPVLab has identified the following expression (RPN) as the solution.

$3 $3 + $5 $4 + + $4 $5 + $1 $9 - / + $3 $4 + $0 $2 + $7 $0 / + + + $1 $4 + $5 $4 - / $3 $6 sin - + $4 $9 sqrt - $9 $1 + sqrt - / +

Note that “$0, $1, $2, $3, $4, $5, $6, $7, $8 and $9” represents 'TR', 'I.CP', 'GGFCE', 'EOG.S', 'OER', 'LEABT', 'PT', 1, 2, 3 respectively. After replacing the real values the RPN expression reads as follows.

EOG.S EOG.S + LEABT OER + + OER LEABT + I.CP 3 - / + EOG.S OER + TR GGFCE + 1 TR / + + + I.CP OER + LEABT OER - / EOG.S PT sin - + OER 3 sqrt - 3 I.CP + sqrt - / +

Then author imported the data into WEKA. After starting WEKA, author selected the Explorer option. Inside the explorer window there is an option to open files. Author selected the same .csv file using this option. Then on the ‘Classify’ tab author selected the ‘LinerRegression’, under functions category, as the classifier. Then under ‘test options’ author selected ‘Use training set’. Finally author selected the ‘(Num) GDP’ as the output attribute. Then author started the process by clicking on the ‘Start’ button. Just after 0.04 seconds WEKA found a solution. WEKA has identified the following as the solution.

GDP = (2.6561 * TR) + (73187869.1757 * I.CP) +
(1083693823.1583 * GGFCE) + (2.8882 * EOG.S) +
(-128547505.6347 * OER) + (966158549.8081 * LEABT) +
(-557.9261 * PT) + (- 65820373354.9022)

After generating models from both systems, user has evaluated these models for each record of that dataset. According to the results the error rate of the solution generated by WEKA falls between -93.74% and 52.00%. Nevertheless the error rate of the solution generated by GPVLab falls between -24.15% and 24.51%. Furthermore in 28 occasions out of 50 records, the expression generated by GPVLab gives more accurate answers than the expression generated by WEKA. That means 56% of the time the expression generated by GPVLab gives better results than the expression generated by WEKA.

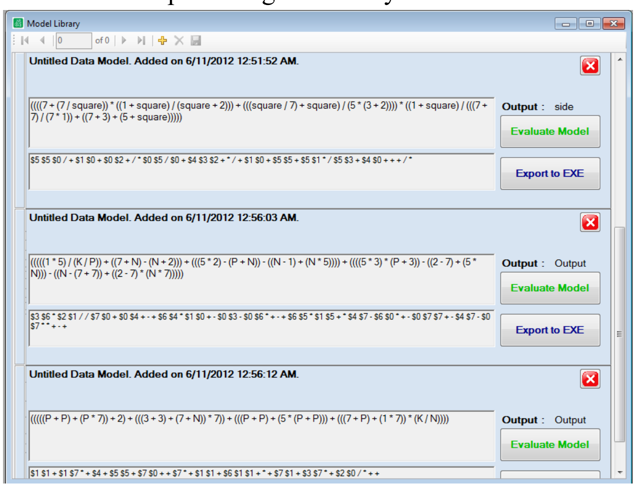

Figure4: –A screenshot of the model library

Furthermore GPVLab has successfully discovered data models in simple known data sets including addition of three numbers, square root of a number and dataset with ten columns with a known data model. All these experiments were conducted by importing the first 80% of the respective dataset into GPVLab. Once the model generation process completed, the resultant model was evaluated for the remaining 20% of the dataset to validate the model. GPVLab has found perfect solutions for the dataset related to addition of three numbers in just 11 generations. The square root of a number experiment has been performed for 5 times with default settings. However, GPVLab has failed to find a perfect solution in all 5 occasions. Then the same experiment has been conducted by changing the function set to “Extended”. This time the GPVLab directly provided the answer using sqrt (square root) function within the first generation. Another experiment has been conducted using a sample dataset with 10 columns. GPVLab has successfully found a perfect in 148 generations. Nevertheless the resultant expression is different than the expected data model.

Another experiment has been done using a noisy real world datasetas well. For this experiment authors have taken “Life expectancy at birth, total (in years)” (LE) and “Official exchange rate (LCU per US$, period average)” (ER) columns from the same world bank dataset. Then authors have imported first 60% of the dataset into GPVLab. Then we have performed series of experiments by changing the function set, maximum number of generations, maximum population and mutation rate. None of these experiments gave a good data model. Therefore we used the noise reduction feature of the GPVLab to find a model for this data set. We were able to find a good data mode after a series of experiments conducted by adjusting the maximum error of an individual. After evaluating the resultant expression for the remaining 40% of the dataset, we saw that this model can approximately predict an answer for a given input. Even though this resultant expression is not a perfect data model, we have found that the margin of error for this resultant expression falls within the range of -2.235 and 3.829.

Furthermore we have conducted various experiments by changing advance settings as well. First, each experiment was conducted using default settings. After recording observations, we performed the same experiment by changing advanced settings. Then we compared the results by the means of accuracy and the number of generations it has taken to find a solution. Furthermore we have given this software to three users who had no knowledge about genetic algorithms or genetic programming. They have successfully adapted to the system and found the system very usable for discovering data models. All three users were happy about the overall system and its capabilities.

8. Conclusion and Further Work

Major aim of this project was to develop a visual environment to design and experiment with genetic programming solutions. This aim has been achieved by developing the GPVLab. According to the main evaluation it is evident that solutions generated through GPVLab provide better results in 56% of the time, compared to the solution generated through conventional linear regression with multiple variables.

Furthermore according to the results obtained through experiments such as addition of three numbers, square root of a number and the experiment with ten column dataset, GPVLab has proved its capability to handle different type of data sets. More importantly the square root of a number experiment showed the importance of 'Advanced Settings' section and the "Extended" function set. The experiment with the noisy real world dataset proved that GPVLab is capable of handling noisy real world datasets. After enabling noise reduction and tweaking noise reduction parameters, system was able to find relatively simple and accurate solutions for noisy data sets. Experiments with advanced settings showed that GPVLab can be used as an advanced tool to experiment with genetic programming problems. After comparing observations collected through numerous experiments, we found that when increasing the maximum population, the system increases the chance of finding a solution within considerably lower number of generations. After going through all the evaluation results we saw that GPVLab has provided some unexpected solutions as well. This proved that GPVLab may lead us to discover a new theory which was previously unknown. At the beginning of the development one of the main objectives is to develop a solution which can be used by non technical persons. For an example a researcher from a non technical domain like agriculture can benefit from this system even without having any knowledge about genetic programming. The results obtained through users with no knowledge about genetic algorithms or genetic programming, proved that this can be a really good tool for the researches in various fields. Considering all these facts it is concluded that GPVLab can be used to model genetic programming application very conveniently.

However, GPVLab is not optimized for large data sets; further work includes optimizing GPVLab to work with large scale datasets. This system may also be extended to take any dataset as an input by encoding non numeric data into numeric representations.

References

[1] J. H. Holland, "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence", 2nd Edition, The MIT Press, 1992.

[2] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", Cambridge, MA: The MIT Press, 1992.

[3] D. A. Augusto and H. J. C. Barbosa, "Symbolic Regression via Genetic Programming", *Proc.of the 6th Brazilian Symposium on Neural Networks (SBRN'00)*, 2000, pp. 173 – 178.

[4] W. Banzhaf, J.R. Koza, C. Ryan, L. Spector and C. Jacob, "Genetic programming", *IEEE Intelligent Systems and their Applications*, vol. 15, no. 3, 2000, pp. 74 – 84.

[5] J. R. Koza, "Darwinian invention and problem solving by means of genetic programming", *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC '99)*, vol. 3, 1999, pp. 604 – 609.

[6] Microsoft Visual Studio 2010, http://www.microsoft.com/visualstudio/en-us, 2012.

[7] Microsoft .NET Framework, http://www.microsoft.com/net, 2012.

[8] AForge.NET Framework, http://www.aforgenet.com/, 2012.

[9] Microsoft SQL Server Compact, http://msdn.microsoft.com/en-us/data/ff687142.aspx, 2012.

[10] D. F. Specht, "A General Regression Neural Network", *IEEE Transactions on Neural Networks*, vol. 2, no. 6, 1991, pp. 568 – 576.

[11] S. Tomioka, S. Nisiyama, and T. Enoto, "Nonlinear Least Square Regression by Adaptive Domain Method with Multiple Genetic Algorithms", *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 1, 2007, pp. 1 – 16.

[12] S. Crino and D.E. Brown, "Global Optimization with Multivariate Adaptive Regression Splines", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 37, no. 2, 2007, pp. 333 – 340.

[13] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", *IBM Journal of Research and Development*, vol. 3, no. 3, 1959, pp. 210 – 229.

[14] R. Balzer, "A 15 Year Perspective on Automatic Programming", *IEEE Transactions on Software Engineering,* vol. SE-11, no. 11, 1985, pp. 1257 – 1268.

[15] N. L. Cramer, "A Representation for the Adaptive Generation of Simple Sequential Programs", *Proc. of an International Conference on Genetic Algorithms and Their Applications (ICGA85)*, 1985, pp. 183-187.

[16] R. Poli, W. B. Langdon and N. F. McPhee, "A Field Guide to Genetic Programming", 2008.

[17] J. R. Koza, "Genetically Breeding Populations of Computer Programs to Solve Problems in Artificial Intelligence", *Proc. of the 2nd International IEEE Conference on Tools for Artificial Intelligence*, 1990, pp. 819 – 827.

[18] H. Tuan-Hao, R. I. McKay, D. Essam, and N. X. Hoai, "Solving Symbolic Regression Problems Using Incremental Evaluation in Genetic Programming", *IEEE Congress on Evolutionary Computation (CEC 2006)*, 2006, pp. 2134 – 2141.

[19] G.Dworman, S.O.Kimbrough, and J.D.Laing, "On Automated Discovery of Models Using Genetic Programming in Game-Theoretic Contexts", *Proc. of the 28th Hawaii International Conference on System Sciences*, vol. 3, 1995, pp. 428 – 438.

[20] GPdotNET, Genetic Programming Tool, http://gpdotnet.codeplex.com/, 2012.

[21] GPLAB, A Genetic Programming Toolbox for MATLAB, http://gplab.sourceforge.net/, 2012.

[22] MathWorks MATLAB, http://www.mathworks.in/products/matlab/, 2012.

[23] GNU Octave, http://www.gnu.org/software/octave/, 2012.

[24] World Bank Data Catalog, http://data.worldbank.org/, 2012.

[25] WEKA (Waikato Environment for Knowledge Analysis), Version 3.6.4, http://www.cs.waikato.ac.nz/~ml/weka/, 2012

Modeling Memory as Conditional Phenomena for aNew Theory of Computing

W. A. C. Weerakoon[1], A. S. Karunananda[2], N. G. J. Dias[3]
[1, 2]Faculty of Information Technology, University of Moratuwa, Sri Lanka.
[3]Faculty of Science, University of Kelaniya, Sri Lanka.
[1]hkcweera@yahoo.com, [2]asoka@itfac.mrt.ac.lk, [3]ngjdias@kln.ac.lk

Abstract **- Computers with the Von-Neumann architecture improve their processing power with the support of memory. This architecture has been improved by introducing different types of memories. The research we conduct has been inspired by the fact that humans are able to improve their memories with the support of continuous processing in the mind. It is evident that we can start with a smaller memory to drive the processing, which in turn improves, both the memory and the processing. This is analogous to how a person uses short-notes to do processing on a larger knowledge base. We postulate the processor to use the said smaller memory to access the bigger knowledge base, without the processor directly accessing the knowledge base as in the present computations. Thus we are researching into the development of the said small memory, say tactics memory, as a novel memory for Von-Neumann architecture. In doing so, we have exploited the Buddhist theory of mind, which presents everything as a phenomenon that occurs when the related conditions are met. We have developed algebra for modeling the tactics memory. The tactics memory can be introduced both as a software or hardware solution for computations.**

Key Words: Theory of computing, Buddhist Theory of mind, 24-Causal Relationships, Functional Programming, Finite Automaton, Memory models, Tactic Memory.

1. Introduction

Von-Neumann(VN) architecture [9]gave birth to a machine that can improve its processing power with the support of the memory. This architecture has been improved by different types of memories such as RAM, Cache and Registers introducing extended architectures [1], such as Harvard architecture and modified Harvard architecture, etc. In recent times, many researchers have been conducted to invent alternate architectures for realizing the computing in novel ways. In this sense, many researchers have attempted to exploit the models of human mind, which are based on different viewpoints about human memory and processing. In these computing models the memory is necessarily distinguished from the processor.

Having the insight from the Buddhist theory of mind [12], we postulate the memory as conditional phenomena which arise due to continuous processing in the mind. According to Buddhist theory of mind [17], everything can be considered as a phenomenon that occurs when the related conditions are met. It is evident that, a smaller memory can initialize the processing, thereby allowing access to a larger knowledge base or a memory of an individual. This scenario is analogous to the use of short-notes to process a huge source of knowledge. The above smaller memory, say tactics memory, also evolves as the processing goes on. We have developed algebra for the tactics memory exploiting the model of 24-causal relations (conditions) presented in Buddhist theory of mind. In implementing the said algebra for the tactics memory, we have extracted 7 causal relations from those 24.

It should be noted that, the tactics memory is not same as the concept of cache or register, because it contains more than the frequently used items. We believe that the concept of tactic memory can extend the VN architecture in a different direction, and contribute to improve performance of computations.

The rest of this paper is structured as follows. The section 2 presents current trends in theory of computing and emerging theories of mind. Then the section 3 discusses the technologies which we can use for the modeling process. The section 4 illustrates our approach for the new computing model. The last section consists of a discussion about the work.

2. Current Trends

This section discusses how the theory of computing has been evolved and its current trends. Further, this section discusses the theory of mind with a particular emphasis on Buddhist theory of mind, and its exploitation towards the development of a novel memory for the VN architecture.

A. Theory of Computing

Theory of computing is a branch of theoretical computer science and it deals with solving problems in an efficient manner using mathematical abstractions which are called as models of computing. Therefore, the researchers have been interested in doing researches on the theory of computing which has the ability to change the direction of the future sciences according to its capability of solving complex problems. As the first computing model, John Von-Neumann presented his computing model taking the insight from the concept "the mill and the store" introduced by Charles Babbage. Still we use this computing model and it consists [9] of addressing, instruction definition, interrupt control and input-output. Other than the VN Computing model, several computing models such as Harvard Models, quantum computing model, register machines, Turing machine etc… have been introduced. Further, Bio-inspired models have been introduced and used in many areas of unconventional computing such as Cellular Automata, DNA Computation; eg: Neural Networks, and associative memory; eg: genetic algorithms. Robustness, adaptability and Scalability can be experienced in these.But, they are lack of unifying methodology, or set of guide lines [16]. These are consists of some of the features similar to the features of VN architecture. Now let us look at few of such computing models in brief.

1) Harvard Architectures

Harvard architecture [1] (see figure 1) consists of two separate buses to read from instruction memory and read/write data from/ to data memory. With this modification both data and instructions can be accessed simultaneously and then it speedup the processing. Even though, this model provides a remedy for VN bottleneck, this model was highly inefficient since, these two busses should be fed.

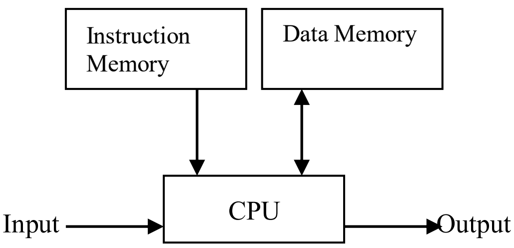

Figure 1: Harvard Architecture

As an improvement, adding a cache [1] in between memory and the CPU in the Harvard model, the Modified Harvard model (see figure 2) was introduced.

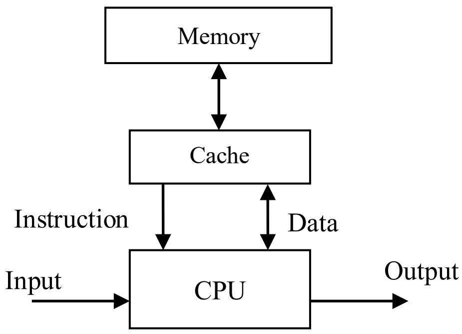

Figure 2: Modified Harvard Architecture

There is, only one pathway to fetch instruction from the memory and read/write data from/to the memory to/from the cache, two pathways to fetch instruction from the cache to CPU and read/ write data from/to the cache to/from CPU. This cache is a kind of a temporary memory, where it contains the data which are currently using or frequently using.

Since these computing models failed in fulfilling the stipulations, researchers developed distributed computing systems [2], which are using clusters consist of the power of thousands or millions of processors. Adding more processors increases the computer capacity of the cluster linearly. According to the Moore's law computer power is doubling for every 18 months by the researchers. But, for each new chip generation, the doubling of capacity means [2] that about half as many atoms are being used per bit of information, then this trend reaches the limit of one atom per bit of information on day. Quantum Computing has the ability on addressing this problem.

2) Quantum Computing

Since, the structure of this consists of components which are called as qubits, with quantum computing, the potential is there to continue, and also to increase the rate of advances in computer problems. Quantum devices eventually become micro- or nano-fabricated and their controls are achieved through the tuning of the laser pulses. A quantum system has "spin states" represented by 0 and 1 [5], respectively called as "spin up" and "spin down" which forms the component which is called as qubit, which is an alternative to the bit. Other than the "spin up" and "spin down" there is a state which is called the "super position", which denotes both the 0 and 1. This super position state can carry large amount of data and it is the difference between the bit and the qubit. These transitions are determined by the selection rules and symmetry of atomic structure. NMR (Nuclear Magnetic Resonance) is the first and the most successful scheme utilized by researches to demonstrate the principle of quantum computing. But it is lack of scalability. SQUID (Superconducting Quantum Interference Device) is hardware and we can see the major progress with that. But the authors' analytical knowledge about SQUIDs still appears to be very limited.

Quantum computers have more capabilities than the classical computers. For example quantum computers can factor large numbers efficiently.There are few limitations [4] over quantum computers; the possibility that speedup in quantum computing can be achieved only for problems which have a few solutions or even a unique solution, the Quantum computers may arrive at results which are fixed-points of a unitary operator or more precisely, Eigen states of unitary or Hermitian operator with Eigen value 1.

3) Turing Machine

Turing machine (TM) is essentially a finite automaton, introduced by Alan Turing, nearly eighty years ago. This has a single tape of infinite length [11] (unlimited memory comes with this) on which it may read and write data. This tape is divided in to cells and these cells can hold any symbol from a finite number of symbols. For this system, the input is a finite-length string of symbols chosen from the input alphabet. TM consists of a finite control and for this finite control there is a pointer which is called as the head, this head scans the symbols in the tape. So then this system can be in any one of finite set of states. Once the machine has finished scanning all the symbols in the string it will be halted.

TM has few extensions, for example Multiple Turing Machines, Nondeterministic Turing Machines, Enumerator, etc… All these similar to TM, with the feature unrestricted access to unlimited memory.In enumerator a printer is attached to it. Since TM is a hypothetical machine, it consists of unlimited and unrestricted memory [11]. Further, TM is more accurate. This has the capabilities of today's computing and it represents the consummate level of today's computing, where the processing is improved by the memory.

4) Computer Memory Models

Memory is an integral part of all models of computing. Therefore, study into alternate models of memories turn out to be a sensible approach to research into come up with novel models for computations.

As mentioned in the section 1, different types of memories such as RAM, Cache, registers were introduced and further modifications has been applying over those from then. Let us look in to some such modifications;

Computation-Centric Memory Models [8]: this theory focus on logical dependencies among the instructions from the way the instructions are mapped into processors and based on the concepts of computation and observer function. The computation specifies machine instructions and dependencies among them and the observer function specifies that for each instance of instruction of computation that reads a value from memory, wrote the value which it is read. As the advantages we can see; since the framework is completely formal, it is easy to show the correctness of the model and this approach can be used to generalize memory models. There they only look in to the formal properties of memory models rather than the implementation of those.

Active Pages [15]: This introduces a computational model which addresses the gap between the processors and memory systems in performance by shifting data-intensive computations to memory system. An active page consists of data and set of functions which can operate on data. Here the computation for an application must be divided or partitioned between the processor and the memory. This provides scalable computations, and optimizes density and indexing. Active pages cannot compute without activation and inter-page communication, both provided by the processor. In this the partitioning process need to be automated and need to minimize the execution time and cost.

Intelligent Co-operative Processor-in-Memory [18]: In this case, the workload is divided between two levels of CPU structure referred to as Major and Minor CPU, in an intelligent manner without having any preprocessor compilation or kernel task scheduling. The Major CPU has conventional architecture and the Minor CPU has a task specific processor dealing with highly iterative memory-to-memory processing. This made more efficient for class of tasks such as image processing algorithms deployed in real time image visualization applications.

It has been acknowledged that it is more important to consider how the memory has been accessed rather than where the memories are located within the architecture of a computer [16]. Moskowitz and Jousselin [14] have also shown that the nature of the operation carried out by a computer processor actually determine the nature of the computer memory. They highlighted the hidden structure of the address space and pointed out that the composition law use by the operation, determines the structure of the memory. This reduces the complexity of computation. Next sections discuss theory of mind with a view to postulate a new model of memory for computers.

B. Theory of Mind

This is a very complicated subject field, where the various explanations are come into view from its researchers in diverse philosophical backgrounds.

To model the mind we need to find invariant and exact mapping between mathematical states and physical states and psychological states [13]. Elodzislaw Duch [6] come up with a similar idea, saying that the problem is in the lack of better language to speak about mind states, and proposing that the cognitive science is on the track of providing a good mind model and such a model should provide a good approximation of brain functions and a connection with psychological states. Even though, these kind of explanations are produced by various researchers, they were failed at providing good explanation for "How the mind works", because there are a lot of challenges in implementing a mind theory. Pentti O. A. Haikonen [8] shows that the problem of identification of the process of mind, meaning, understanding, qualia, emotions, consciousness and inner speech. Because of the theories of mind are introduced by the philosophers those who don't have any technical knowledge, the ability to implement these theories, is not easy as the engineers are not provided much guidance from these explanations.

What the engineers need is a clear specification [8] for the system design; in other words they need some algorithmic and computational approach for the expected function using commonly accepted engineering terms. Further, Haikonen [8] arouse that when we try to simulate such a theory, the particular simulation won't cover the entire theory. For this, as the reason he claims that when doing such simulations, simplifications and shortcuts should be called to have reasonable time for the processing. Finally, he says that even though these issues can be addressed using high level algorithmic approach and low-level system approach. But then, these approaches have their own issues to be addressed.

Together with these challenges, psychologists have been introducing models for the memory and then the scientists have been trying in building computational models using those memory models. Now, Let us discuss about such prominent memory models.

1) Human Memory Models

The first model [23] of human memory was “the boxes in the head” which was introduced by Waugh & Norman (1960) which consists of short term store and long term store. Then the “Classic information-processing model of Memory” was introduced in 1968. In the second model there is an extra memory to store the memories at the moment the information sensed, which is called as sensory memory. In the both cases short term memory occurred due to stimulus or information will pass to the long term memory only if they maintained the rehearsal over the short term memories.

Zhang [24] modifies the above second model, calling the dynamic view of short term memory as working memory and dividing the long term memory in to two; procedural and declarative memory. He says the declarative memory stores all the knowledge that can be consciously accessed or expressed symbolically through speech or writing and procedural memory stores all the skilled performance such as typing and riding. He proposes that as in the second model once a sensor gets information, at that moment, the particular information is stored in the particular sensor memory and then passed to the short term memory until sleeping time. If the human being is in the sleep, the information stored in the short term memory is transferred in to the long term memory. Again as the classes of information stored in long term memory; special information, physical laws, beliefs, values and social goals, motor skills and perceptual skills are introduced.

Reducing mental life to computational [22] procedures led to move from cognitivism to connectionism. There the connectionism [22] has common points of view with the Buddhism, such as consideration of a behavior as resulting from relationships.In past twenty years, new optimism about the relevance of Buddhism [7] to cognitive science has been seriously considered by a growing number of established researchers such as James Austin (1999; 2006), Richard Davidson (2008), Christopher deCharms (1997), Daniel Goleman (1991; 1997), Jeremy Hayward (1987), Eleanor Rosch (2007), Francisco Varela (1991) and Alan Wallace (2003; 2007; 2009). From the scientific point of view, Buddhism brings extra value, and from the ethical point of view it brings empirical credibility. If we move to the Buddhist theory of mind, it provides a comprehensive theory of human mind which undertakes so many cognitive tasks such as thinking, reasoning and remembering.

2) Buddhist Theory of Mind

Buddhist theory of mind defines the mind as a continuous flow of thoughts that arise when the conditions are met. The mind involves in non-ending processing. According to Buddhism everything can be considered as a conditional phenomenon. So does the mind. The thoughts in the flow of thought are characterized by what we refer to as mental factors. There are 52 mental factors such as perception, volition, anger, jealous, attachment and mindfulness. The mental factors also arise when the conditions are met. Among the fifty two mental factors, mindfulness is the closet connotations to the concept of memory in our terms. Thus memory can also be treated as a conditional phenomenon. In other words, formation of memory can be explained through the study of conditions that can manifest any phenomenon in the universe.

Buddhism has identified 24 causal relations [17] or the conditions to describe any phenomena in the universe. They are: Root (Hetu), Object (Arammana), Predominance (Adhipathi), Proximity (Aantara), Contiguity (Samanantara), Co-nascence (Sahajatha), Mutuality (Annamanna), Support (Nissaya), Decisive-support (Upanissaya), Pre-nascence (Purejata), Post-nascence (Pachchajata), Frequency (Asevana), Karma, Karma-result (Vipaka), Nutriment (Ahara), Faculty (Indriya), Jhana (Jhana), Path (magga), Association (Sampayutta), Disassociation (Vippayutta), Presence (Atthi), Absence (natthi), Disappearance (Vigata), and Non-disappearance (Avigata).

Some of these conditions may not be directly perceived in layman terms as they refers to aspects related to ultimate truth as well. However, some have the usual meaning. For example, arammana or objects work as a condition for formation of thoughts. This means, visual objects condition the visual perceptions. So do for all other five senses. As another example, disappearance works as a condition for something to happen by terminating an ongoing thing. This can be found in processor controlling in computing domain. Pre-nascence is also one of conditions, which stands for having happened something is requirement for some other thing to happen. In the computing domain, this concept can be seen in execution of procedures with multiple functions or procedures.

Here, we do not intend to describe all the causal relations one by one. Soon we will explain the selected causal relations or the conditions that are relevant for modeling the special kind of memory that we stated above.

C. Inspiration for a new model of memory

On the way towards a novel model for computing, we have the insight from the following phenomena which we can see in the real world.

Things we process more are better established in the memory and can be processed faster. Based on this concept, people uses a short notes to access the whole knowledge base. When a student prepares for an examination this kind of memory can be improved by doing tutorial, past papers, and some ‘tactics’ for answering questions. Some teachers who conduct revision classes are very successful in developing the memory of students. This is the memory that operates on the knowledge base to drive answers in an examination. Without such memory students cannot answer question effectively, regardless of richness of the knowledge base. All skill workers have their own such memory which in fact exemplifies their level of competency. Experienced lecturers conduct lectures without any notes as they have well structured and updated such memories.

It is evident that human beings are excellent in execution of programs through the above mentioned small memory. However, this phenomenon has not yet been exploited to develop software or hardware strategies to improve performance of computers.

3. Technologies to Model Process

With the basis provided by the Buddhist theory of mind, now we have to think of technologies modeling memories. As Buddhism defines memory as result of processing, the technologies that can be used to model the processing can also be used to model memories. Since the functional programming has the ability to model processes and the theory of finite automaton has ability to simulate computing models, we can consider these as the technologies to model the intended new memory. Therefore, let us identify the use of these technologies.

A. Functional Programming

Functional Programming (FP) [21] is one of the programming paradigms. The beneficial way to look in to these programming paradigms is; how they model real world or imaginary situations. In Functional Programming the major concern is on the relationships of values and FP comes with a central component of model which is called as "Function" that gives an output depending on the input. These input values or object should be of certain type and the function manipulates or processes these values or objects. Then, this Function concept can be used to model the computing model. This allows us to feed the intermediate results of the process back to the ongoing process, while giving them as the output.

B. Theory of Finite Automaton

Finite Automata (FA) is simpler kind of machine, which was originally proposed [11] to model the brain. Now FA can be used in many areas other than this; to check behavior of digital circuits, as a lexical analyzer of a typical compiler, to scan large bodies of text to find occurrences of phrases or words, to verify systems which of those consist of finite number of distinct states.

If we consider any system or component, these are in any one of the finite number of discrete states at a given time. These states have the responsibility in maintaining the memory of systems history. Since there are only finite numbers of states, this remembering all these is not possible. Therefore the system should be designed carefully with fixed set of resources.

Since this deals [19] with definitions and properties of mathematical models of computation, this is an excellent place to study of the theory of computation. Therefore, this can be used to simulate the new memory model of our research.

4. Novel Approach to Computing

This section introduces the tactics memory for computers and discusses a new approach for the theory behind the tactics memory for the performance improvement in the execution of computer programs.

A. Tactics Memory

We postulate the existence of a small memory that has the power to explore the whole knowledge base or the main memory. We name this memory as the *tactics memory*. Further, by its very nature, the tactic memory grows in size and updates over the time. More importantly, the tactics memory holds a collection of strategies to execute a huge program in the main memory. We believe that tactics memory could be implemented as a software solution as well as a hardware solution.

Here an element in the tactics memory triggers in the presence of related conditions as per the main program, on a similar way to the continuous process of improving the human memory. Further, the execution of a program is nothing but a result of satisfying the conditions in the body of a procedure. This goes with the Buddhist philosophical thought that everything in the world can be explained as conditional phenomena. As explained earlier, since the 24-causal relationships can model any process in the world, we propose to use a subset of causal relationships, in modeling the tactics memory.

B. Exploiting causal relations from Buddhism

We exploit the causal relations or conditions defined in Buddhism for a theory of new kind of memory with the reference to the domain of computing. We also use a real world example to illustrate our concept.

At the outset, quite intuitively we treat any process as a union of sub processes. We are very much impressed by the condition known as co-nascence (sahajatha), which states the processing of related things together. Based on this processing, the human mind forms its entire processing. Our rationale of exploitation of Buddhist theory of mind can also be illustrated with a real world example as follows.

Let us think about the cooking process in the kitchen. There are a lot of sub processes such as washing, peeling, cutting vegetables, get seasoned with spices, scrape coconut and make coconut milk. Here, we do all the similar or related things together like washing, peeling, cutting all the vegetables together, get seasoned all together. In this instance, we can see the causal relationship Co-nascence and also there are repeated occurrences. Automatically, related things are mapped in to the clusters washing, peeling, cutting vegetables, get seasoned, make coconut milk and these things we can concern as the fundamentals. We accomplish these tasks one after the other, taking the result of one to the next. Then one task should be completed and finished before start the next task. The causal relationships absence or disappearance can be seen here.

Being based on the co-nascence conditions, following section presents the formal model of the tactics memory.

C. Formal Model of the Tactics Memory

Let us define the tactics memory more formally. The tactics memory is a 3-tuple: $<I, P, C>$, where

- I - Set of inputs
- P - Set of processes
- C – Set of clusters
- $\exists P_C \in P$, such that $P_C(I) \rightarrow C_k \in C$
  $\exists x, y \in C_k, x \perp_{relate} y$ or $x \perp_{not\ relate} y$
  $k = 1, 2, \ldots, n$
- $\forall P_j \in P$, such that $P_j (C_j) \rightarrow O_j, j = 1, 2, \ldots., n$
  (the output of P_j)
- $\exists P_\lambda \in P$, such that $P_\lambda(C_k) \rightarrow \Phi$
  (the null output)

Architecture of the tactics memory is shown in Figure 3.

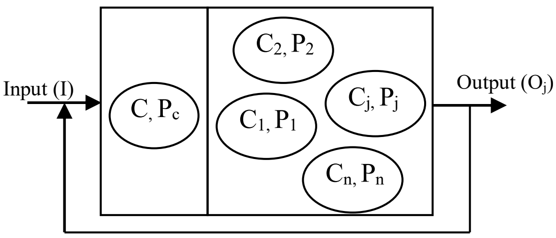

Figure 3 - Architecture of Tactics memory

D. Implementation of the Tactics memory

This section presents an algorithm and gives a description of the implementation of the tactics memory.

Here also we can see the set of states such as new (created), waiting, ready, running, exit (terminated) which the processes in operating systems [20] are occupied.

Algorithm:

- This model consists of a buffer (C) which contains set of clusters, and within this three kinds of processes such as P_C, P_j and P_λ can be taken place.

- Initially, C is empty and three processes are created and idling. (These processes are in the "new" state)

- At the presence of an input $i \in I$;

  - If and only if an ongoing process is ceased (the ongoing process can be in the state "exit" where the process is aborted or fully completed), the input i can be accepted.

  - If the input i is accepted;

(Now P_C is at the "ready" state and other two processes P_j and P_λ become "waiting")

- P_C maps $i$into $C_k(k=1, 2, \ldots, n)$ clusters in C according to a fundamental criteria. (Now P_C is at the "running" state and other two processes P_j and P_λ still in "waiting")

  - There, the n-1clusters consist of related things ($\exists x, y \in C_k, x \perp_{relate} y$) and one cluster consists of non-related things($\exists x, y \in C_k, x \perp_{notrelate} y$).

  (After finishing this mapping process P_C is in the "exit" state and other two processes P_j and P_λ become "ready")

  - Depending on i;
    - C_j is processed through P_j (P_j is in "running" state, P_C is in the "waiting" state, P_λ is in the "ready" state), producing the result O_j. (After P_j become "exit", other two processes P_C and P_λ become "ready")

    Or

    - C_j is processed through P_λ (P_λ is in "running" state, P_C is in the "waiting" state, P_j is in the "ready" state), producing no output Φ. (After P_λ become "exit", other two processes P_C and P_j become "ready")
    $j=1,2,\ldots, n$

    - Result of one process P_j, $O_j(j=1, 2,\ldots,n)$ can become an input $i \in I$ to the next process, while giving the result O_j $(j=1, 2,\ldots,n)$ out.

In order to give the above definition, a sub set of 24-causal relations is exploited as said earlier. This subset consists of 7 causal relations (conditions) such as Root, Pre-nascence, Co-nascence, Absence or Disappearance Frequency and Happening (Karma-Vipaka).

Table 1 shows the conditions that we have exploited in modeling the tactics memory.

Table 1 – Conditions exploited from Buddhism

Condition	Purpose
Root	Define the clusters on the basis of fundamentals of a domain to implement related things
Pre-nascence	An ongoing process conditions the formation of a new process.
Co-nascence	Related things in a cluster are process together
Absence	A process can cease after termination
Disappearance	A process can cease forcefully before terminate
Frequency	repeated occurrence
Happening	Something happening with or without being related to what happens now. (Karma-vipaka)

Let us discuss how we have exploited the above mentioned set of conditions in defining the tactics memory.The above definition present a module of clusters of related things based on the condition Co-nascence to process related things together. This module is attached with three processes; one (P_C) to create clusters, next (P_j) to execute clusters one after the other producing outputs and the last (P_λ) to execute cluster without producing an output. In creating the clusters (C), the condition Root is exploited to set the fundamental criteria which are then used by the process P_C to classify inputs in to clusters. The process $P_C$conditions the formation of a new process, i.e. process P_j or the process P_λ which process the clusters one after the other as the conditions meet. Here the condition Pre-nascence is exploited. If the process P_C or P_j or P_λ completes and ceases its process, then it is absent. If the process P_C or P_j or P_λ is forcefully stopped, then it is disappeared. As the conditions meet, a clustercan be processedor such a process can be occurred with or without being related to current process. Karma-vipaka condition can be seen there. If the same input is received for multiple occurrences then the processes related to that should occur multiple times. There we can see these processes are frequently occurring.
As above, the subset of 24-causal relations has been exploited in implementing the new model.

5. Discussion

This paper presented a new theory for computation. We postulated memory as a conditional phenomenon exploiting the model of 24-causal relations in Buddhist theory of mind. New memory named as tactics memory which consists of set of strategies those capable of executing complex programs in the main memory has been proposed. A subset of relations extracted from the above mentioned 24-causal relations have been used to define the tactics memory. The causal relation *co-nascence,* processing the clusters of related things together provides the basis for the entire work. Finally a formal model of the tactics memory has been presented with the details of its implementation. It is pointed out that tactics memory can be implemented both as software or a hardware solution. At present we are in the process of implementing and doing experiments over the tactics memory as a software solution towards a new theory for computation. We believe that with this model we will be able to improve the performance of the computer.

References

[1] Abd-El-Barr M., El-Rewini H. (2005) *Fundamentals of Computer Organization and Architecture*, John Wiley & Sons, Inc., Hoboken, New Jersey.

[2] Black P. E., khun D. R. and Williams C. J., (2002) *Quantum Computing and Communication*, Advances in Computers, Academic Press, vol. 56, pages 189-244.

[3] Burgess N., Hitch G. (2005), *Computational models of working memory*: putting long-term memory into context, TRENDS in cognitive Sciences, Vol.9 No.11 November 2005.

[4] Calude C. S., Michel J. D. (2000), *Reflections on Quantum Computing*, CDMTCS (Centric For Discrete Mthematics & Theoretical Computer Science) Research Report Series-130.

[5] Chen G., Church D. A., Englert B. G. and Zuhainy M. S. (2003) *Mathematical models of Contemporary Elementary Quantum Computing Devices*, CRM Proceedings.

[6] Duch W. (1998), *What constitutes a good theory of mind?*, American Neopragmatism, Torun 8-12.08.

[7] Federman A., (2011) *What Buddhism Taught Cognitive Science about Self, Mind and Brain*, Quardens de Filasofia 47, pp 39-62.

[8] Frigo M., Luchangco V., (1998) *Computation-Centric Memory models*, SPAA'98 proceedings of the 10th Annual ACM symposium on parallel algorithms & architectures, pp 240-249, ACM, New York, NY, USA.

[9] Godfrey M. D., Hendry D. F. (1993), *The Compputer as Von-Neumann Planned It*, IEEE

Annuals of the History of Computing, Vol 15, No 1, 1993.

[10] Haikonen P. O. A., (2008) *The Challenges for implementable theories of mind*, Journal of Mind Theory. Vol. 0, No1. pp 99 – 110.

[11] Hopcroft J. E., Motwani R., Ullman J. D. (2001), *Introduction to Automata theory, Languages and Computation* 2nd edition, Pearson Education.

[12] Karunanda A. S. (2002) *Using an eastern philosophy for providing a theoretical basis for some Heuristics used in artificial neural networks*. Malayasian Journal of Computer Science, Vol. 15 No. 2, December 2002, pp. 28-33.

[13] Litch M., (1997) *Computation, Connectionism and Modeling the Mind*, Philosophical Psychology, Sep 97, Vol. 10 Issue 3, p357, 8p.

[14] Moskowitz J. P., and Jousselin, C. (1989), *An Algebraic Memory Model*, ACM SIGARCH Computer Architecture News Archive, 17(1), 55-62, ACM Press, New York, NY, USA.

[15] Oskin M., Chang F. T., Sherwood T., (1998) Active *Pages: A computation Model for Intelligent Memory*, Proceedings of 25th International Symposium on Computer Architecture, Barcelona.

[16] Propenko M, Wand P. (2008), *Optimizing Associative Information Transfer within Content-Addressable Memory*, vol 4(3), Special issue "Towards Theory of Unconventional Computing", 273-296.

[17] Rathnapala A. (1964) *Abhidharmartha Pradeepika*. Vol 3. Free Distribution, http://www.aathaapi.org.

[18] Reza S., Ahmad Z., Bensaali F., (2007) *Intelligent Co-operative Processor-in-Memory*, Mediterranean Journal of Electronics and Communications.

[19] Sipser M. (2006), *Introduction to the Theory of Computing* 2nd Edition, Thomson Corse Technology, Thonson Learning Inc.

[20] Stalling W. (2012), *Operating Systems, Internal and Design Principles*, 7th Edition, Prentice Hall, India.

[21] Thompson S., (1999) *Haskell, the Craft of Functional Programming*, 2nd Edition, Addison-Wesley.

[22] Vallverdu J., (2011) *The Eastern Construction of the Artificial Mind*, Quardens de Filasofia 47, 2011, pp 171-185.

[23] Wells J. E., *Some Current Trends in Memory based on North American works,* Massey University.

[24] Zhang J. (2004), *Memory Process and the Function of sleep*, Journal of Theoretics, Volume 6-6, December2004.

Monocular Vision Based Agents for Navigation in Stochastic Environments

P.A.P.R Athukorala[1], Asoka S. Karunananda[2]
[1,2]Faculty of Information Technology, University of Moratuwa, Sri Lanka
[1]paprathukorala@gmail.com, [2]asoka@itfac.mt.ac.lk

Abstract- **Autonomous navigation in a stochastic environment using monocular vision algorithms is a challenging task. This requires generation of depth information related to various obstacles in a changing environment. Since these algorithms depend on specific environment constraints, it is required to employee several such algorithms and select the best algorithm according to the present environment. As such modeling of monocular vision based algorithms for navigation in stochastic environments into low-end smart computing devices turns out to be a research challenge. This paper discusses a novel approach to integrate several monocular vision algorithms and to select the best algorithm among them according to the current environment conditions based on environment sensitive Software Agents. The system is implemented on an Android based mobile phone and given a sample scenario, it was able to gain a 66.6% improvement of detecting obstacles than using a single monocular vision algorithm. The CPU load was reduced by 10% when the depth perception algorithms were implemented as environment sensitive agents, in contrast to running them as separate algorithms in different threads.**

Keywords**— Software agents, Monocular vision, optical flow, appearance variation.**

1. Introduction

Depth perceiving computer vision algorithms which are based on multiple view geometry are computationally expensive. As such, it is not practical to implement such systems in low end computing devices such as mobile phones. Nevertheless, for certain applications, monocular computer vision based algorithms which are capable of generating depth approximations are adequate and can be implemented on low end computing devices. In this context, we are still faced with the problem that monocular vision is very much affected by environment conditions such as light intensity, noise, density of obstacles, depth, etc. In case of stochastic environments, these aspects are even more crucial. Accuracy of each algorithm depends on its internal constraints and environment conditions which that particular algorithm is capable of handling. For that matter, it is required to execute multiple monocular vision algorithms in a system and to select the result of the most appropriate algorithm according to the current environment condition. As such modeling of monocular vision based algorithms for navigation in stochastic environments into low-end smart computing devices turns out to be a research challenge.

One approach to autonomous navigation from monocular vision is to use machine learning techniques [1]. There are other methods based on interesting points [15], feature pairs [16] and defocus [12], which are mostly based on mathematical models constructed using mechanical and imaging properties of the system. Among the mentioned approaches, Machine learning based approach is capable of integrating several depth perception techniques to derive a depth map of the environment.

Our research to address the above issue postulates that the Agent technology can model such environment sensitive situations. By definition, an agent is a small program that autonomously activates when necessary, performs a task and terminates on the completion of the task. This amounts to optimize the resource usage, which is a crucial factor for low-end computing devices. On the other hand Agents can negotiate and deliver high quality solutions which go beyond the individual agent's capacity to solve a problem. Also Agents are reactive to their environment and they can make decisions according to changes in the environment.

This paper is organized as follows. Section 2 describes various monocular depth perception techniques used by computer vision based navigation systems. Section 3 contains the technology adapted and section 4 contains our novel approach based on agent technology to solve the problem. Section 5 contains more detail on designing monocular vision based agents as environment sensitive software agents. Section 6 contains the implementation of the system and section 7 contains the experimental results and finally, the conclusion and further works is presented in Section 8.

2. Related Work in Monocular Vision Based Navigation and Depth Perception

There exist different techniques based on different types of sensors to navigate stochastic environments, such as IR, Ultrasonic and Vision. In most systems, the environment is reconstructed based on the data observed by these sensors, where the reconstructed 3D model is used to generate navigation decisions. One major advantage of selecting a vision sensor over others is that it can be easily used to extract some additional information about the environment, such as identifying the type and color information of obstacles, identify human faces and so on. Furthermore, the vision sensors are cheap, versatile and can be used with learning algorithms to improve over time. A comprehensive comparison among vision and IR sensors for depth perception is given in [8]. In our system, we are using a single vision sensor to make navigation decisions.

Vision based autonomous navigation is a vastly studied subject among the researchers in computer vision and robotics. According to DeSouza et al. [3] two major areas of vision based navigation exist, as indoor and outdoor navigation. Indoor navigation can be further classified based on map-based, map building or maples navigation strategies. Approaches for outdoor navigation can be based on structured or unstructured environment conditions.

The system developed by Pan et el. [7] is one of the earlier systems developed for autonomous indoor navigation based on fuzzy logic and an ensemble of neural networks. Task of the ensemble of neural networks is to generate a sequence of basic steering commands based on topological models of hallways generated using the indoor environment. The ambiguities inherently associated with these interpretations of steering commands have been dealt using fuzzy logic. Each steering command is treated as a command with a certain ambiguity associated with it and a fuzzy logic based controller provides higher-level of intelligent control over these steering. This approach points out one important aspect of vision based sensors that require attention, which is the inherent ambiguity in vision based sensors. This system is designed only for an indoor environment and the algorithms which are being used to generate navigation commands are fixed. In addition, it uses a sonar system and does not make any decisions based on the vision sensor pertaining to obstacle avoidance.

The Generalized feature vector [4] method developed by J. Bhattacharya et al. can be used to improve the accuracy of vision based outdoor navigation and is resilient to the extrinsic parametric variations of interested objects. They highlight the drawbacks of relying on only one feature to identify the objects and use multiple features organized in to a feature vector. This concept also aligns with our approach, where the design of the system can accommodate different feature detection algorithms.

Apart from the technology and design perspectives, another important aspect of vision based navigation is the underlying depth perception techniques which are being used with these systems. It is interesting to observe that some of these techniques are based on different aspects of human vision system. According to Schwartz [10] and Loomis [6], humans relay on four major visual cues to perceive depth. They are namely monocular, stereo, motion parallax and focus cues. Monocular cues provide depth information when viewing a scene from one eye. This includes relative size, color, texture variations and lightning information. The concept of visual cues has been used to generate the 3D depth map by Saxena et al. Their approach [1] is based on machine learning and contains a large training set of monocular images and their correspondent ground truth depth maps. In the training phase, a Markov Random Field has been used to predict the value of the depth map as a function of image. The algorithm combines several image cues with some previous knowledge to generate the depth map. Although it is capable of generating visually realistic depth maps form a single 2D image, their approach does not mention on generation of depth information from a real time video sequence, which is essential for an Autonomous navigation system.

A general domain independent tool [2] for automatic discovery of depth estimation algorithms has been developed by C. Martin. His work is based on Genetic algorithms and is capable of generating depth perception algorithms according to domain specific constraints such as the relationships between the various obstacles in a given environment. Although the evolved program has produced promising results, it requires a supervised learning framework and has to be trained against a pre-existing environment. One important aspect of his work is that it points out the importance of generating domain specific depth perception algorithms in order to handle various complexities in stochastic environments.

X. Lin and H. Wei have developed a method [15] based on the displacement of an interested point in an image sequence. This method does not require any prior knowledge of the image sequence and only depends on the focal length of the camera. Their approach is based on perspective transformations, by which the three dimensional world coordinates are projected in to two dimensional camera coordinates. Since the inverse of such perspective transformations does not support the generation of depth values directly, they have used multiple images to generate a sequence of image projection planes and introduced a novel mathematical equation based on the focal length of the camera to calculate depth information of selected feature points. The algorithm requires keeping track of the interesting objects in the scene across multiple images, which is done by a matching method based on brightness of the object. The algorithm is easy to be implemented in a real time system and it exhibits a comparatively good accuracy according to the given experimental results. But in an environment where point matching is not possible, it is difficult to generate depth estimations using this approach. For an example, when the autonomous navigation unit is in front of a plain colored wall, it might not be possible to detect any feature point.

The "Hypothesize-and-Test" approach [16] proposed by Y. Fujii et al. requires the knowledge of approximate displacements of the robot along the focal-axis of the moving camera. The algorithm hypothesizes that there is a pair of feature points having the same depth and does its calculations. As the camera moves, the depth map is built depending on the validity of the hypothesis. This approach is better suited for a slow moving robot equipped with other mechanical sensors to measure its relative position. Generation of the depth map is an iterative process which progresses with the motion of the robot and the complexity of the algorithm prevents it from using with fast moving robots and low end mobile devices. This approach also fails when there are no feature points to be located.

R. Kumar et al. have proposed a method [9] to automatically identify the 3D locations of image features from a sequence of monocular images captured by a mobile camera. The algorithm is having two steps as to build an approximated shallow 3D model and a refined 3D model based on the shallow structure. The shallow structures, as defined by [11] are structures whose extent in depth is small compared to their distance from the camera. Affine transformations [12] are being used to generate these shallow structures. Although the method is capable of generating more realistic results, it is difficult to be used in a real time system equipped with a single camera due to the fact that it requires the same object to be captured in many different angles.

V. Leroy et al. [13] have constructed a mathematical model to represent the relationship between different blur levels and the depth of an image object. This technique is widely known as “depth from focusing”. Based on the Gauss law of the thin lenses, they have constructed a mathematical equation which relates the optical properties and blur level of the lens with the depth of the observed objects. In order to be success with the algorithm, it is necessary to capture the same object using at least two different focus settings. Experiments have shown the mean error for the algorithm as 7%. If machine learning techniques have been incorporated in to the algorithm, it would have been possible to overcome the most errors originated due to noises. Also there is a possibility of integrating fuzzy logic in to the decision making process of this algorithm. Drawback of this approach is that it requires the same object to be captured using several blur levels and difficult to be used in real time navigation systems.

J. Cardillo and A. Sid-Ahmed [5] have also used the concept of depth from focusing to generate the absolute 3D coordinates of the objects from their observed camera coordinates. Although they have achieved Position accuracies comparable to those in stereo vision systems, the system requires calibration and the calculations have a dependency with sharp edges appear in the image.

Among the algorithms and navigation systems we have discussed, a clear separation of two classes of approaches can be noticed. One approach is based on visual cues and machine learning techniques, which is capable of accommodating more than one depth perception algorithm, handling noises and can adapt to changes in the environment. But these algorithms depend heavily on training data and as per the complexity of image processing is concerned; a large set of training data is needed. Other approach is based on constructing a mathematical model with the help of mechanical properties of the system. This approach provides comparatively accurate results, but lacks noise handling and adoptability on stochastic environments. It was also noticed that none of these approaches has much concern about integrating awareness of its environment, a crucial factor which decides applicability of an algorithm on a particular environment.

3. Technology Adapted

Software agent technology is a new paradigm to model distributed systems. It consists of multiple autonomous agents having the same or different goals to achieve. They are decentralized and can work in parallel to each other. As opposed to software objects, agents do not run code on demand of others, but decides for itself to perform some activity. Communication among agents happens through passing messages to each other. Message passing enables agents to perceive the current state of the system and update its decision making process accordingly. Agents have to use a common language to communicate each other and ACL is such a language introduced by FIPA [14].

Software agents exhibit flexible behaviors. They are reactive to their environment and are capable of making decisions according to what it perceive at a given instance. Due to this nature, agents are more robust, flexible and fault tolerant than conventional software programs. In a stochastic environment, a reactive agent is capable of adapting the changes quickly. Agents also exhibit a proactive nature by having a self initiated execution behavior in situations, rather than waiting for someone to request to do some task. They can work with minimum supervision and does not need in detailed instructions.

We have adopted the request-resource-message-ontology architecture to build the system, which is shown in Figure1. Ontology is the formal representation of knowledge used in a particular domain. The relationships among various concepts are also built in to the Ontology. In a Multi agent system, Ontology can be any source of knowledge in any format such as a Database, website or even a text file. Two agents can successfully communicate only if they have a shareable Ontology. Also the learning process of an agent is the process of updating and editing its Ontology.

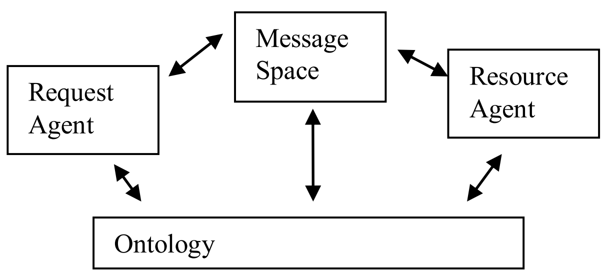

Figure1: Request-resource-message-ontology architecture for MAS

The system contains three request agents, namely, Appearance variation based agent, Optical flow based agent and Floor detection based agent. These three agents represent three unique depth perception algorithms. Hardware agent is the only resource agent present in the system, which is responsible for acquiring and sending necessary image frames from the mobile phone camera to request agents.

4. Agent's Navigation in Stocastic Environments

Our approach is based on modeling several monocular vision algorithms as environment sensitive software Agents. Each agent in the system represents a unique depth perception algorithm and is reactive to the environment at present time. When a particular environment is not in favor for a particular Agent, it does not continue with the depth estimation process and tries to minimize its update cycles by allowing other Agents having a better confidence on that environment to update more frequently. Agents in the system are autonomous and it is Agent's responsibility to define its confidence and execution frequency on a particular environment. Final depth estimation value is selected according to the most confident agent in the given environment. This approach improves the overall accuracy of depth perception in a stochastic environment by being able to select the best algorithm according to changing environment conditions, while minimizing the resource requirements. Furthermore, particular outcome from the system in a given instance is not predetermined and is emerged based on the most confident Agent at that moment.

5. Design of the System

As shown in Figure2, current design contains a Hardware agent, three depth perception agents and a message space agent.

This architecture is highly extensible and allows several depth estimation processes to run in parallel as separate agents, while enabling communication among them. Each Agent in the system can be a simple computer vision based algorithm or can even represent a total different technology, such as a Machine learning process.

The hardware agent initiates the camera of the device and inputs an image to the system for the use by appearance variation based agent, floor detection based agent and optical flow based agent. The message space agent displays the communication and enables negotiations among agents. Appearance variation based agent, Floor detection based agent and optical flow based agent have small codes to represent unique monocular vision based algorithms which are capable of generating depth approximations to various obstacles.

As shown in Figure 3, Design of the Optical flow agent requires two consecutive images and a list of detected feature points. Lucas–Kanade optical flow calculation has been used to calculate the optical flow. After calculation of the optical flow vectors, a time to collide calculation is conducted and if the time to collide is less than a defined threshold value, it classifies that particular vector as an obstacle which is going to collide. Center of the image is taken as the point of expansion during these calculations.

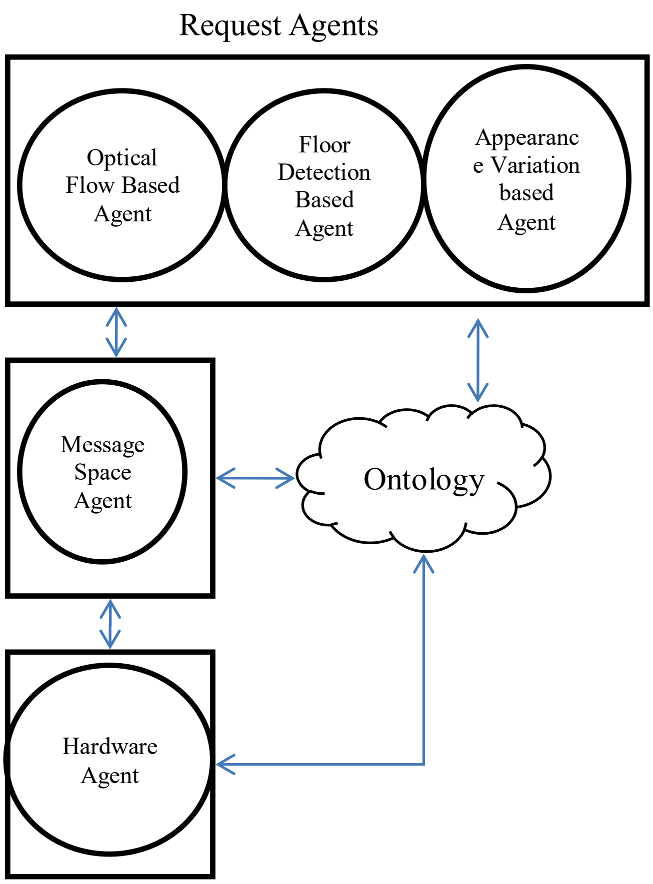

Figure2: High level design of the system

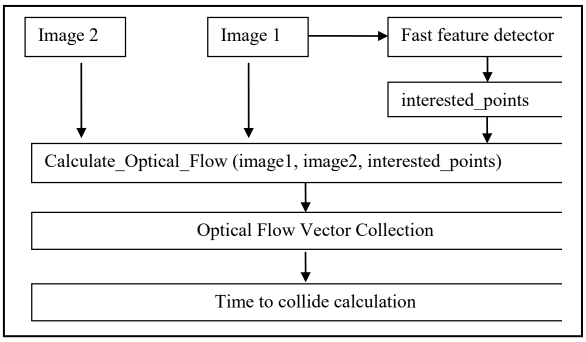

Figure3: Design of Optical flow calculation

Appearance variation for a particular image is calculated using the Claude Shannon's theory of information, which deals with encoding large quantities of information. As shown in figure 4, when agent receives an image, it converts it in to a gray-scaled image, which is an optimization technique where we get a chance to bypass all the color space details.

Thereafter, the probability distribution of the occurrence of gray levels is calculated. Finally the

Shannon entropy is calculated based on the calculated probability distribution. A smaller entropy value represents a smaller distribution of gray levels and hence, the image is assumed to be an obstacle.

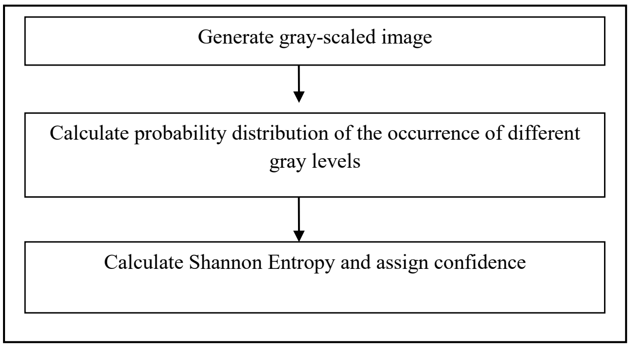

Figure4: The appearance variation calculation

Reason behind to select an appearance variation agent and an optical flow agent is that they work well in two different environments. For the Optical flow Agent, it is required to track some feature points from the input image sequence and its prediction is based on the flow of these points. In an environment where feature points are difficult to track, this agent cannot be used. In other terms, when the appearance variation of the environment is low, optical flow agent does not work well. In contrast, appearance variation agent requires the environment to be less in variation, which is the indication of a nearby obstacle. However, it should be noted that there can be conflicting situations where a detectable set of feature points are still available in an environment where the appearance variation is low. Floor detection based agent is another important agent which only activates when it finds that the camera is facing towards the floor of the environment. In such situations, floor detection based agent should get the priority among others and it is capable of detecting any obstacles lying on the floor.

Confidence value and the execution frequency of the optical flow agent are directly proportional to the gradient magnitude of the input image. In other words, an image which has lot of detectable edges is required for the optical flow agent. The appearance variation agent's execution frequency and confidence values are inversely proportional to the calculated Shannon entropy. This is due to fact that when the variation of appearance is high in a particular environment, appearance variation agent is not capable of indicating any nearby obstacle. Confidence value and execution frequency for the floor detection based agent is directly proportional to the orientation of the camera. When the camera is directly facing down, its confidence reaches the maximum value.

6. Implementation of Agents

The system is implemented on an Android based mobile phone having a 1GHz processor and a 512 MB RAM.

Agent frame work is implemented with the help of inbuilt messaging and threading routines of the Android platform.

The OpenCV image processing library is used to implement the image processing algorithms. Pseudo code for the implemented optical flow agent is presented in Figure 5.We are using the Lucas–Kanade optical flow estimation technique, which is a widely used differential method for optical flow estimation. Feature point detection is based on the Fast feature detector. Also Figure 6 and Figure 7represents pseudo codes for the implemented appearance variation and floor detection based agents respectively.

```
IF(IsConfidentEnough() )
{
        CreateGrayScaledImage();
        ChangeColourSpaceSuitableForOpenCV();
        DetectFeaturePoints();
        calculateOpticalFlow();
        calculateTimeToCollide();
        SendMessageToMessageSpaceAgent()
}
```

Figure 5: Pseudo code for the optical flow based agent

```
IF(ConfidentEnoughToRunThisCycle())
{
        CalculateHistogram();
        CalculateShanonEntrophy();
        classifyAsObstacle();
}
```

Figure6: Pseudo code for the appearance variation based agent

Major difference between appearance variation and floor detection based agents is in their confidence evaluation strategies. Appearance variation based agent uses the calculated Shannon entropy to measure the confidence, while the floor detection based agent is using the camera angle.

```
EvaluateConfidenceUsingCameraAngle();
IF(ConfidenceEnoughToRunThisCycle())
{
        CalculateHistogram();
        CalculateShanonEntrophy();
        classifyAsObstacle();
}
```

Figure 7: Pseudo code for the floor detection based agent

7. Experimental Results

Experiments were conducted in a sample environment to evaluate the agents sensitiveness to the environment, the system's ability to improve the decision making process in a stochastic environment and the system's resource utilization.

Given a stochastic environment, implemented agents are capable of detecting Continuous changes in the environment and to redefine their confidence levels accordingly. At the same time, the agents are capable of adjusting their execution frequencies according to the environment. This ability is tested by moving the camera towards a selected sample object in the living room. As shown in figure 8, at the initial position where the obstacle is far away from the camera, the optical flow agent has a better confidence than the appearance variation agent. Optical flow agent has a confidence of 96 % and all the other agents are having a confidence of 50%. This is due to the feature rich nature of the given environment. When the camera is getting closer, the execution rate of the appearance variation agent also increased. This is shown in Figure 9 where the optical flow agent is having a confidence of 96% and the appearance variation agent is having a confidence of 75%.

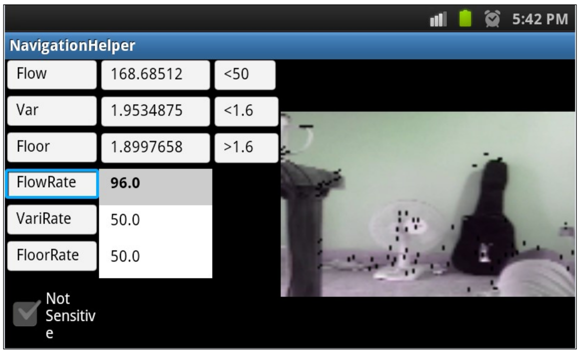

Figure8: Confidence of agents when obstacles are away from camera.

Figure9: Confidence of agents when the camera is getting closer to an obstacle.

When the camera image is covered with the obstacle, the appearance variation was the selected agent for making depth estimations because the appearance variation of the image becomes extremely low. This situation is shown in Figure 10. Since the obstacle was not on the floor, the floor detection based agent did not provide any depth estimations with a higher confidence level throughout the experiment, but immediately activates when the camera is pointed towards the floor.

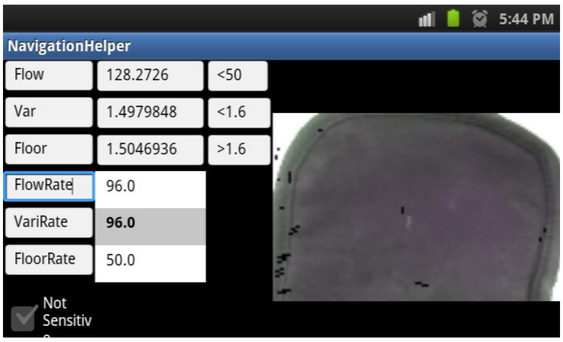

Figure10: Confidence of agents when the camera is near by to the obstacle.

In order to improve the decision making process in a stochastic environment, at least one agent should be able to generate results with a higher confidence when exposed to different environment conditions. Three experimental scenarios were setup to evaluate this objective. In the first scenario, the camera was held against a plain colored wall, where it is difficult to find feature points to track. In this situation, the optical flow agent failed to detect any obstacles. Also the floor detection agent was able to distinguish it from a plain color floor and did not exhibit a higher confidence. As shown in Figure 11, this scenario was successfully handled by the appearance variation agent by detecting the wall with a confidence value of 75%.

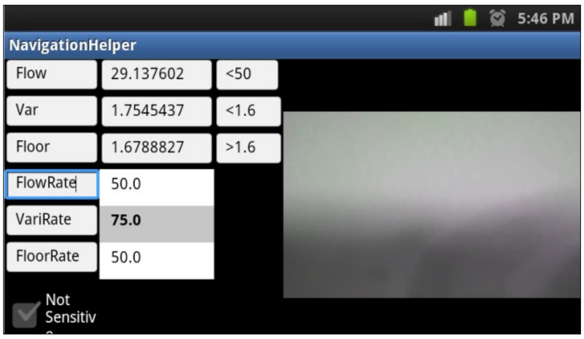

Figure 11: Confidence of agents when camera is pointed towards a wall.

In the second scenario the camera was pointed towards a colorful obstacle. This situation is shown in figure 12. Confidence values of the appearance variation based agent and the floor detection based agent remained at a lower level due to the large variation of gray levels, but the optical flow agent was capable of detecting enough feature points and executed with a confidence of 96%.

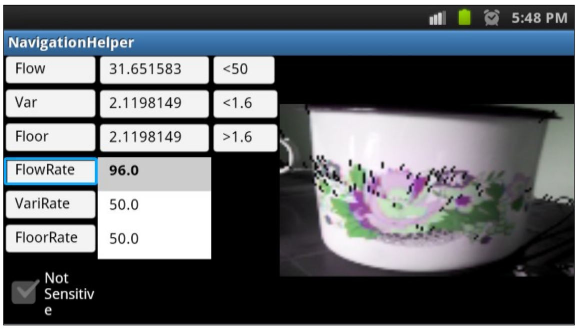

Figure12: Confidence of agents when the camera is pointed towards a colorful obstacle.

In the third scenario, the camera was pointed directly towards the floor of the environment. In this environment, the floor detection based agent gets the priority over the others by executing with a Confidence of 96%, which shown in figure 13.

According to evaluation results on the sample stochastic environment, the system has displayed a 66.6% improvement of detecting obstacles than using a single monocular vision algorithm. Since the depth perception process is contributed by the most confident agent on a particular environment, this kind of system definitely improves the decision making process of a navigation system.

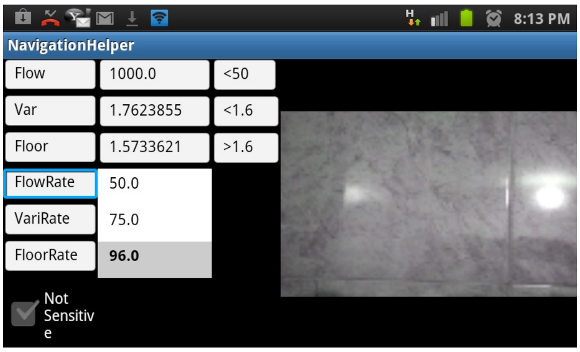

Figure13:Obstacle on the floor is detected by the floor detection agent

Whenmultipleimageprocessingalgorithmsarerunninginasystem,itisessentialtoallocatememoryandprocessingpoweroptimallyamongthesealgorithms.Inthedevelopedsystem,agentsdonotutilizeresourcesatallthetime. Whentheenvironmentisnotinfavorforthem,theydo not execute any depth estimation calculations and also reducetheirupdatefrequencies.Bydoingsotheseagentssavememoryandprocessorcyclesofthesystem.As shown in shown in Figure 14 and Figure 15, the CPU load has been reduced by 10% when the depth perception algorithms are implemented as environment sensitive agents, in contrast to running them as separate algorithms in different threads.

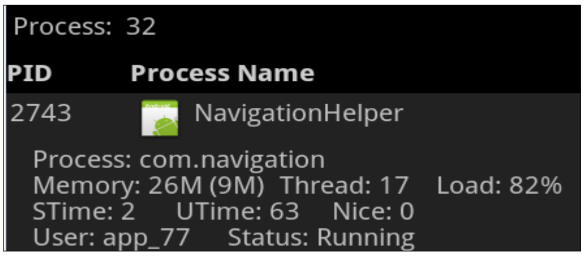

Figure14: Memory and processor statistics when the agents executing at full speed.

This clearly indicates a reduction in processor usage in the agent based environment sensitive version. But due to the caching mechanisms used in OpenCV and Android operating system, statistics of the memory usage could not be obtained in a reliable manner.

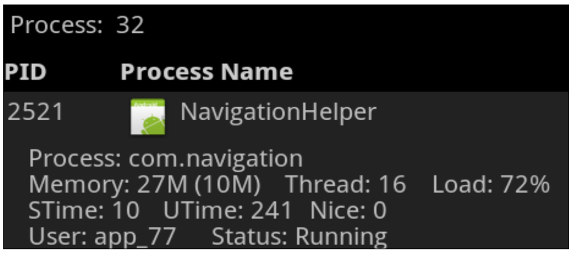

Figure15: Memory and processor statistics when the agents are sensitive to the environment.

8. Conclusion and Further Work

In this paper, we have presented a novel approach for monocular vision based navigation based on Multi Agent Technology. We have modeled several depth perception algorithms in to environment sensitive software agents. As per the evaluation results, a clear improvement has been achieved in resource utilization and depth perception. Improving the mechanism to determine the confidence of an agent by an automated machine learning process is one of the major further works. It is possible to go through a machine learning process to identify the environments where the agent is more confident. Agent's reaction for a given environment has to be based on this machine learning process. This is a complex task and the training process should cover adequate environments which could occur in day to day life.

References

[1] A. Saxena, M. Sun and Y. Andrew, "3-D Depth Reconstruction from a Single Still Image", *International Journal of Computer Vision (IJCV)*, vol. 76, no 1, January 2008.

[2] C. Martin, "Evolving Visual Sonar: Depth From Monocular Images", *Pattern Recognition Letters*, vol. 27, 2006

[3] G. N. DeSouza and A. C. Kak. "Vision for mobile robot navigation: A survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, February 2002.

[4] J. Bhattacharya, S.Majumder, "The Generalized Feature Vector (GFV) : A New Approach for Vision Based Navigation of Outdoor Mobile Robot", *Proc. 14th National Conference on Machines and Mechanisms (NaCoMM09)*, NIT, Durgapur, India, December, 2009

[5] J. Cardillo and A. Sid-Ahmed, "3-D position sensing using a passive monocular vision system*", IEEE transactions on pattern analysis and machine intelligence*, vol. 13 no 8, August 1991.

[6] J. M Loomis, "Looking down is looking up", *Nature News and Views*, 2001, pp. 155–156.

[7] J. Pan, D.J. Pack, A. Kosaka, and A.C. Kak, "FUZZY-NAV: A Vision-Based Robot Navigation Architecture Using Fuzzy Inference for Uncertainty-Reasoning," *Proc. IEEE World Congress Neural Networks*, vol. 2, July 1995, pp. 602-607.

[8] P. Viswanathan, J. Boger, J. Hoey and A. Mihailidis, "A Comparison of Stereovision and Infrared as Sensors for an Anti-Collision Powered Wheelchair for Older Adults with Cognitive Impairments", *Proc. 2nd International Conference on Technology and Aging (ICTA)*, Toronto, 2007.

[9] R. Kumar, S. Sawhney and R. Hanson, "3D model acquisition from monocular image sequences", *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992.

[10] S. H. Schwartz, "Visual perception, a clinical orientation", McGraw Hill Professional, 2004

[11] S. Sawhney and R. Hanson, "Identification and 3D description of 'shallow' environmental structure in a sequence of images", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992, pp. 179-186

[12] S. Sawhney and R. Hanson, "Affine Trackability aids Obstacle Detection", *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 1992, pp. 418 – 424

[13] V. Leroy, T. Simon and F. Deschênes, "An efficient method for monocular depth from defocus", *Proc. 50th International Symposium(ELMAR)*, IEEE Computer Society, 2008, pp. 133 – 136

[14] The foundation for intelligent physical agents, FIPA Specifications, Available at: http://www.fipa.org/repository/aclspecs.html

[15] X. Lin and H. Wei, "The Depth Estimate of Interesting Points from Monocular Vision", *Proc. International Conference on Artificial Intelligence and Computational Intelligence(AICI 2009)*, IEEE Computer Society, 2009, pp. 190-195

[16] Y. Fujii, K. Wehe, E. Weymouth, "Robust Monocular Depth Perception Using Feature Pairs and Approximate Motion", *Proc. IEEE International Conference on Robotics and Automation*, IEEE Computer Society, 1992, pp. 33 – 39

Price Prediction Neural Network for Econ Centre

Prasad Wanigasinghe[1], D D M Ranasinghe[2]

Department of Electrical & Computer Engineering
The Open University of Sri Lanka,
1visionx.prasad@gmail.com[1], ddran@ou.ac.lk[2]

Abstract - **Econ centre price prediction system works as a business intelligence system that practically helps wholesalers to predict the high marginal profit selling price of an item. According to the survey done based on "Dambulla" economic centre, increase or decrease of one rupee of an item can cause one wholesaler to gain or loss around 5000 rupees per day. Minimizing the loss by selecting the high marginal profit selling price is the main aim of this system. This aim was achieved by combining neural network time series analysis techniques and financial analysis techniques in cost accounting. This system is able to predict the required variables down to 1.01% error percentage. Using those predicted values the system then calculates the high marginal profit selling price and alternative prices.**

1. Introduction

Business intelligence plays a vital role in a developing country like Sri Lanka. Economic centers can be identified as one of the major business sectors in economy. But due to lack of knowledge, lack of sponsors, and lack of advisors, combining technology and theory into daily business needs is still at a minimum level at Sri Lankan economic centers. Still most of the wholesalers use traditional methods such as guessing and factors like relationships, bonds and agreements between producers and traders to decide selling price, although there is high competition.

Most of the times due to lack of use of technologies, wholesalers cannot gain whatever they are able to gain or sometimes they may lose their business.

Increase or decrease in one rupee for a particular item (for 1 Kg) can be a factor to gain or loose profit at the end of the day for wholesaler. Not only that, if he gains profit, sometimes it may not be the highest profit that he can gain. For example, think about the situation where one wholesaler has good quality items and customers also have a trust on his items. Then for this particular wholesaler, the best price may not be the price of the other wholesalers because his products are of high quality. Therefore he can sell for a higher price. In case if the wholesaler sells at the same price as of other sellers, at the end of the day he may not gain the higher value that can actually gained from that item. Consider another example, where another wholesaler has an item that is of not that high quality as of others. But if he tries to sell them to other wholesaler's prices it can be a cause for him to loose.

In Dambulla Econ Centre average trading of items are 26500 Metric Tons per week [4]. That implies, for one day, average of 3786 Metric Tons (3786000 Kg) of items is being sold. Let's assume there are 1000 wholesalers in Dambulla Econ Center. Then one wholesaler sells around 3786 Kg per day. Suppose that the average unit price of an item is 100 rupees, and then income for one day is 378600 rupees. If the average unit price of that item is 101 rupees, then income for a day is 382386 rupees with a 3786 rupee difference.

Therefore the problem is that how can a wholesaler determine the most benefited price of an item. To address this problem a price prediction system is being developed using neural networks.

The rest of this paper is organized as follows. Section 2 contains a literature survey done on existing similar systems. Section 3 describes the methodology and design of this system. Section 4 describes the implementation of the project. Section 5 is about results and discussion and the final section 6 is about conclusion and future works.

2. Literature Survey

This system is developed to address the problem of predicting high marginal profit selling prices at economic centers in Sri Lanka. The project is developed as a business intelligence system that practically helps wholesalers.

Some similar technologies to predict price were studied like technical analysis, fundamental analysis, traditional time series forecasting, Chaos theory and other techniques like fuzzy logic and expert systems. Technical analysis based on the assumption that history repeats itself and that future market direction can be determined by examining past variations. Price movement analysis in speculative market [13] is an example of prediction using technical analysis. Fundamental analysis, as in earning prediction [14], involves the in-depth analysis of a performance and profitability to determine its item price [6].

Time series forecasting analyzes as in passenger analysis [15] past data and projects estimates of future data values. In chaos theory, chaotic system is a combination of a deterministic and a random process. The deterministic process can be characterized using regression fitting, while the random process can be characterized by statistical parameters of a distribution function. Example for Chaos is market reality analysis [16]. Each and every approach has its own pros and cons. The major benefit of using a neural network then is for the network to learn how to use above mentioned other approaches for prediction, in combination effectively and hopefully learn how the market behaves [6] [9].

Stock price prediction system [17] and house price prediction system [18]are two similar systems being studied. The ability of neural networks to discover nonlinear relationships in input data makes them ideal for modeling nonlinear dynamic systems such as the stock market. Also for the house price prediction, comparing to traditional prediction methods, Neural Network based price prediction provides a reasonable error minimized prediction. Neural networks are relatively easy to implement compared to other methods. Most important benefit is the ability to learn from data and thus produce an acceptable output for previously unseen input. This even holds input series contain low-quality or missing data. Furthermore no expert system is needed which makes the network extremely flexible to changes in the environment[2].

3. Methodology and Design

From the literature survey the technologies adopted in similar systems was identified. In order to identify the day to day factors that affect the selling price of an item a questionnaire was carried out at the Dambulla economic center. About 50 wholesalers participated in this survey. Figure 1shows the results of the questionnaire in percentages.

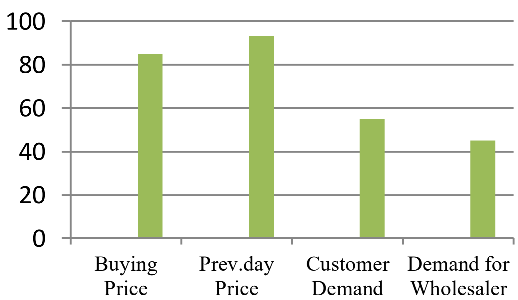

Figure1: Questionnaire results

According to the results wholesalers mainly consider about the buying price, last price of an item, customer demand to that item and demand for the wholesaler.

All the above tasks were done to recognize what are the main factors that really affect the item price. As identified, this business environment can take as a perfect competitive market. According to theory [3], [8] some factors like, reasonable price offering should attract more orders, Quality-Consumers are likely to prefer better quality items, a buyer may look elsewhere if a firm cannot deliver quickly and on time, an image through various methods of marketing are some other factors that affect selling price. Combining these factors together with the factors identified from the questionnaire, buying price, demand to the item, demand to the wholesaler and last price are used to define input equations to the Neural Network. The overall system design is given in below figure 2.

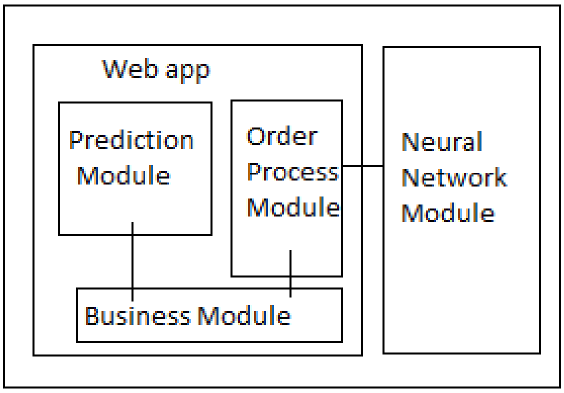

Figure2: Overall System Design

Implemented system consists of four sub modules, prediction module, business model development module, order processing module and neural network development module. The first three modules are integrated with each other and developed as a web application. While neural network development module, specially developed to implement, train and compare neural networks, was built as a separate standalone system.

Web application was developed to capture required data from both buyers and wholesaler (shop owner) and to address some practical issues related to buyers. It facilitates buyers to place orders; view prices etc through online and minimize unnecessary costs and time wasting.

Business model development module was build to develop an equation to determine the high profitable price and to determine other factors such as alternative solutions, calculate the movement of selling/buying prices and selling/ buying quantities.

Neural network development module was built to identify the best suitable structure for the networks. Neuroph, which is a low level Java based neural network library, was used to develop and train the required neural networks.

4. Implementation

Implementation was done in three steps. First necessary equations were implemented to determine high marginal profit price. Then neural networks were implemented to predict necessary data and then web system was implemented to glue whole system together.

Financial equations development

Financial equations development is focusing on developing models/ equations to determine high profit marginable selling price. Since the market at economic centers can be classified as highly competitive, cost accounting techniques can be used to calculate the high profit marginable selling price by using the general equation [11],

Selling price = cost / (1-GP) (1)

Where GP is the gross profit

Determination of cost and gross profit for above equation totally depends on the situation. As found in the survey, the factors that affect the selling price are; previous selling prices and quantities for last five or six days, purchasing prices, quantities for last five or six days, weather situation and customer demand.
According to the available data and factors determined from the survey, four main variables are selected as that affect equation 1. The selected variables are, selling price, selling quantity, purchasing price and the purchasing quantity. These variables were considered for the last fifteen (15) months.

The actual cost was calculated by adding fixed cost to variable cost. Here purchasing cost is a fixed cost for a particular day. At the end of the day, purchasing price of remaining quantity, which is calculated by reducing selling quantity by purchasing quantity and multiplied by purchase price, is the variable cost. So, for the above equation, cost was calculated as below by considering the following abbreviations,

HMCSP-High Marginal Cost Selling Price
PP- Purchasing Price
SQ – Selling Quantity
PQ-Purchasing Quantity
SP-Selling Price

Cost = PP + [(PQ – SQ)/SQ] x PP (2)

To calculate gross profit, selling price and purchasing price were used.

GP = SP – PP (3)

Then the whole equation would be,

HMCSP = {PP + [(PQ – SQ)/SQ] x (PP)}/ [1 - (SP – PP/100)] (4)

Using equation (4) alone can calculate only high profit marginable selling price for the previous data. In other words, using only the above equations the selling price for next day cannot be found. To do so, equation variables should be related to next date. To find those, neural network is used.

Neural network development

Neural Networks are used to predict, variable values needed by (4) to calculate high marginal profit selling price for the next day. The following four variables are predicted by the Neural Networks.

Selling price for next day – what will be the price that seller is going to sell particular item next day
Selling quantity for next day – what will be the quantity that seller is able to sell next day
Purchasing price for next day – what will be the price seller will buy that particular item next day
Purchasing quantity for next day – how much quantity seller will buy next day

Those four variables are predicted using four neural networks. Neural networks take previous buying and selling data as inputs, then process it and predict the output. This includes some stages as indicated below,

The neural network is trained in two stages. In stage one previous 20 day relevant variables (previous 20 day selling price for first Neural Network and so on) is taken as the input.
In stage two data is pre-processed to make them ready to input to the neural network. According to the activation function, data has to be kept between 1 and 0. In doing so, two other variables, max value and min value are being defined. The highest value in the data set; max value is multiplied by 1.2 and the lowest value; min value is multiplied by 0.8 to avoid small 0.00 values and big 0.999 values (values 1.2 and 0.8 have chosen to get max/min value in equation (5) away from 0 but close to its value). Then input data are processed by,

Input data [i] = Data[i] – min / max (5)
i – relative data number

The above pre-processed data, input into neural networks and get the predicted output.

In next stage, predicted values are input to do financial analysis and calculate the high marginal profit selling price.
Not only given the output, in next stages altogether generates graphs to determine the variable movements (selling / purchasing price movement, selling / purchasing quantity movement). Then analyze graphs and produce best and alternative solutions.

Training

Since there is no theory to determine best suitable structure for any neural network the suitable structure was identified by experimenting and changing structure several times. Following table summarize the different structures and relevant error percentages.

Table 1 – Summary of Error percentages

Input nodes	Hidden nodes	Number of data input (days)	Input iterations	Error percentage
5	11	20	100	3.46%
5	11	50	100	2.20%
5	11	50	1000	2.14%
5	11	70	1200	2.44%
5	11	70	1500	2.86%
10	21	20	100	2.84%
10	21	50	100	2.37%
10	21	50	1000	1.31%
10	21	70	1200	1.57%
10	21	70	1500	2.01%
20	41	20	100	2.02%
20	41	50	100	1.53%
20	41	50	1000	1.01%
20	41	70	1200	1.61%
20	41	70	1500	1.84%
30	61	20	100	1.38%
30	61	50	100	1.20%
30	61	50	1000	2.45%
30	61	70	1200	2.43%

Final neural network structure contains 20 inputs, 41 hidden nodes and 1 output. Input to the particular network would be the previous 20 day data. Training has done using 200 days inputs and iterates it 1000 times. Following diagram illustrates how the network is trained.

Neural Networks get data from previous available data for that particular variable (i.e. sold prices for carrot, sold quantities for carrot, purchased prices for carrot, purchased quantities for carrot).

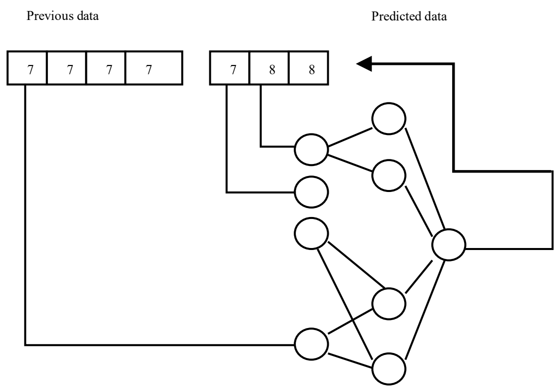

Figure 3 – Training Neural Network

If consider only one variable (say, purchased quantities for carrot), for one input, 20 data from that variable has taken. Then they are pre-processed to keep between 1 and 0 as described in previous section. To do training and developing neural networks, Neuroph [12] was used.

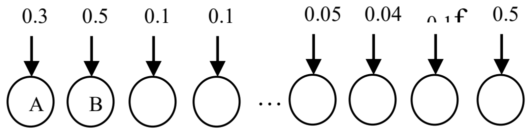

Figure 4 – Initial input assign

Neuroph

Neuroph is an object-oriented neural network framework written in Java. It can be used to create and train neural networks in Java programs. Neuroph provides Java class library as well as GUI tool "easyNeurons" for creating and training neural networks. [12]

It is the open source project hosted at Source Forge, and the latest version 2.4 has been released under the Apache License. Previous versions were licensed under LGPL.

5. Results and Discussion

To check the accuracy of the neural network, one year period of data taken by the economic center was used. First, those data were divided into two sets as training data and testing data. Using training data, neural network was trained. After that, results generated by neural network compared with testing data.

Below example demonstrate how neural network predict values and how to use that result to calculate high marginable profit selling price. The diagram below shows the interface used to insert above configurations to train the neural network. During the training time, configurations were changed frequently to select the best structure for the neural networks.

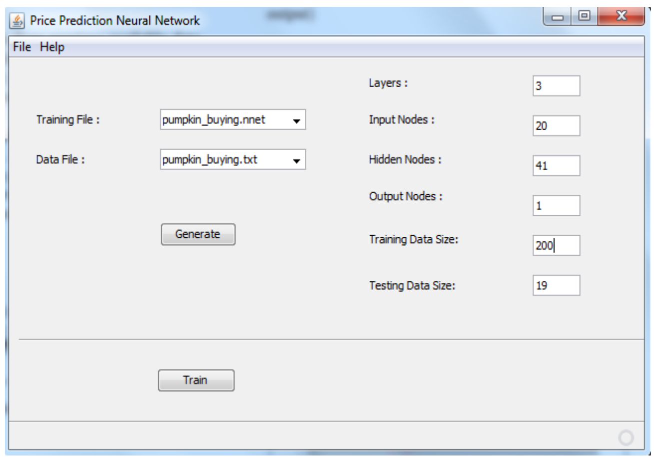

Figure 5 – Modifying Neural Network

Target: Predict buying price for pumpkin
Number of layers: 3
Input nodes: 20
Hidden nodes: 41

Training data size: 200 (which is the number of inputs to train the network)
Testing data size: 19 (which is, after training the network, number of actual data that compare with predicted output)

After trained the neural network, predicted data was compared with testing data as in the figure below.

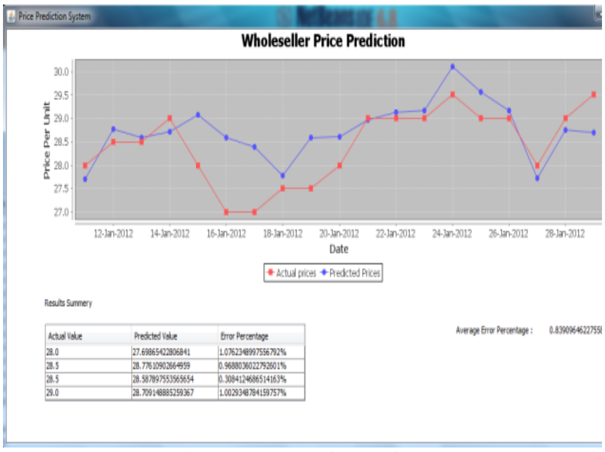

Figure 6 – Sample result

According to figure 6, there are some situations where fluctuations of predicted data are very high. Those are the situations that external factors like weather situations, government decisions affect to real price, which is beyond system predictions.

Then using above predicted value and other predicted values (selling price, buying quantity and selling quantity for pumpkins), high marginable profit selling price is being calculated as below.

Table 2 – Sample predicted results

Variable	Predicted Value	Actual Value	Error %
Predicted buying price for next day	Rs. 75.45	Rs. 75.00	0.6 %
Predicted buying quantity for next day	21.51Kg	22 Kg	2.22 %
Predicted selling price for next day	Rs. 80.35	Rs. 81.00	0.8 %
Predicted selling quantity for next day	19.21 Kg	19.5 Kg	1.48 %
High marginal profit selling price	Rs. 85.36		
Predicted buying price for day after next day	Rs. 76.30	Rs. 77.00	0.9 %
Predicted buying quntity for day after next day	21.71 Kg	23 Kg	5.6 %
Predicted selling price for day after next day	Rs. 81.40	Rs. 83.00	1.9 %
Predicted selling quantity for day after next day	20.06 Kg	19.5 Kg	2.87 %
High marginal profit selling price (Assuming to sell remaining items in day after next day)	Rs. 83.45		

6. Conclusion and future works

This system is being developed to predict high marginable profit selling price of an item. System consists of financial analysis part to calculate the best selling price and neural network part to predict future values. Neural network was developed using Neuroph, which is a low level java library to develop neural networks.

Up to now, for a single item there are four neural networks operating to predict four variables.

Neural network 1 – predict purchasing price
Neural network 2 – predict purchasing quantity
Neural network 3 – predict selling price
Neural network 4 – predict selling quantity

The main reason to have four neural networks is so that the seller will be able to analyze movement of each variable for the next day. However there are two alternatives available to reduce number of neural networks used.

One alternative is, first calculate the high marginable profit price for each previous day and take those as input to neural network. The main advantage of this method is there is only one neural network for single item. But seller cannot analyze the individual variable movements, which is the disadvantage of this alternative.

The other alternative is building a larger neural network using bias values to predict more than one variable within the same neural network. Therefore number of neural networks used can be dramatically reduced. Also seller will be able to analyze individual variables as well. But the problem in this approach is it requires to build a huge (take around 80 input data) neural network that is suitable for the available data. Otherwise data can be overlapped. And also, accuracy will be low comparing to individual neural networks.

The correctness of results depends primarily on predicted output by neural network and the accuracy of the equations developed. However this implemented system provides a solution up to some extents and is able to predict values down to 1.01% error percentage.

References

[1] Jacobson, Ivar, Unified Modeling Language User Guide, 2nd Edition 2005, Addison-Wesley Professional

[2] Rich, Elaine,(2005),Artificial Intelligence. 3rd Edition. McGraw-Hill publications, New Delhi

[3] Anderton, Alain,(2005), Pearson Education. 3rd Edition. Singapore

[4] Ministry of Co-operate and internal trade web site. http://www.giclk.info/trade/ [19.08.2010]

[5] GledenGem Software web site. http://www.goldengem.co.uk/ [11.11.2010]

[6] Ecosystem Valuation web site. http://www.ecosystemvaluation.org/hedonic_pricing.htm [18.12.2010]

[7] High Beam Research web site. http://www.highbeam.com/doc/1G1-145683239.html [03.11.2010]

[8] investopedia web site. http://www.investopedia.com/university/technical/techanalysis1.asp [13.11.2010]

[9] investopedia web site. http://www.investopedia.com/terms/f/fundamentalanalysis.asp [06.11.2010]

[10] DTREG tool web site. http://www.dtreg.com/TimeSeries.htm [05.12.2010]

[11] Horngren, Charles. Cost Accounting, 14th Edition, Pearson publications.

[12] Neuroph web site.http://neuroph.sourceforge.net/ [08.12.2011]

[13] Alexander, S.S., Price movement in speculative markets: Trend or random walks, No. 2, in Cootner, P.

[14] DynaSeng. Fundamental Analysis and the Prediction of Earnings.Department of Accountancy and FinanceSchool of Business, University of Otago.

[15] DTREG tool web site. http://www.dtreg.com/TimeSeries.htm [09.12.2011]

[16] Reality based training company web site. http://www.rb-trading.com/article4.html [09.12.2011]

[17] BirgulEgeli,Stock Market Prediction Using Artificial Neural Networks. Asst. Prof.

Bogazici University, HisarKampus, Istanbul, Turkey.

[18] Limsombunchai. Price Prediction:Hedonic Price Model vs. Artificial Neural Network. Commerce Division, Lincoln University, Canterbury 8150, New Zealand.House.

Organizing Committee

- Dr. L.S.K. Udugama, The Open University of Sri Lanka
- Dr. (Ms) D.D.M. Ranasinghe, The Open University of Sri Lanka
- Dr. (Ms) H.U.W. Ratnayake, The Open University of Sri Lanka
- Prof. A. S. Karunananda, University of Moratuwa
- Prof. Nalin Wickramarachchi, University of Moratuwa
- Ms. Chandrika Fernando, Sri Lanka Institute of Information Technology
- Mr. Budditha Hettige, University of Sri Jayewardenepura
- Ms. P.T.R. Dabare, The Open University of Sri Lanka
- Ms. S.N. Dissanayake, The Open University of Sri Lanka
- Mr. D.S. Degalla, University of Peradeniya
- Mr. Ramesh Maddegoda, Virtusa (Pvt) Ltd

Panel of Reviewers

- Prof. A.S. Karunananda, University of Moratuwa
- Prof. N. G. J. Dias, University of Kelaniya
- Prof. Nalin Wickramarachchi, University of Moratuwa
- Dr. (Ms) D.D.M. Ranasinghe, The Open University of Sri Lanka
- Dr. L.S.K. Udugama, The Open University of Sri Lanka
- Dr. A.P. Madurapperuma, The Open University of Sri Lanka
- Dr. Prasad Wimalaratne, University of Colombo School of Computing

Technical Session Chairs

- Prof. R.G.N. Meegama, University of Sri Jayewardenepura
- Prof. Gihan Dias, University of Moratuwa

Panel of Judges– Best paper Award

- Dr. A.P. Madurapperuma, The Open University of Sri Lanka
- Dr. T.G.I. Fernando, University of Sri Jayewardenepura
- Dr. L. Ranatunga, University of Moratuwa