

Agent-Based Knowledge Representation for Computational Problem Solving

M.A.Teh Noranis, and N. Shahrin Azuan

Abstract—Computational problem solving visualization tools lack information on the techniques to guide novice students before writing the source code. Problem solving understanding is vital to avoid misconceptions of the computer programming theory and incorrect program output. Due to these problems, a novel model based on agent is invented to aid novices in computational problem solving. The proposed agent model is based on the standard Belief, Desire and Intention (BDI) model. Knowledge representation are denoted by conventional problem solving techniques consisting of the Problem Analysis Chart (PAC), Hierarchy Input Process Output (HIPO) chart, Input Process Output (IPO) chart, flowchart and algorithm. These techniques are based on agents that provide a complete step-by-step explanation for novice reference during source code writing. Two modules namely the Problem Solving Comprehension Module and the Assignment Module are designed. The main role of agents in the Problem Solving Comprehension Module is to extract information, transform one problem solving techniques to another and generate module number. This module consists of six agents namely GUI, PAC, HIPO, IPO, flowchart and algorithm agents. The main role of agents in the Assignment Module is interaction with the user. This module consists of four agents namely Editor, Novice, Assessment and Guidance agents. The model is tested with three different problems and proved that the proposed algorithm worked accurately.

Index Terms—agents, knowledge representation, problem solving, visualization

1 INTRODUCTION

Problem solving techniques is one of the earliest topics taught to novice students in the fundamental computer programming course and extra time is allocated to teach the topic. Understanding a problem clearly is very important for novices before writing the actual program source codes. However, based on experience, most of these students find it very difficult to understand the problem given and transform the problem into problem solving techniques consisting of PAC, HIPO, IPO, flowchart and algorithm. Problems that are misinterpreted by novices may cause an incorrect result and misconceptions of the programming concepts. This may also lead to programming errors and bugs. Previous researches have focused on various techniques to enhance the comprehension of novices in computational problem solving [1]-[3].

Software agents have the potential to facilitate in the learning and teaching process. Agents have been widely applied in various applications such as interactive tutoring [4], medical diagnosis system [5] and image analysis [6]. Featuring four properties namely autonomy, pro-activeness, reactivity and social ability, agents have assisted humans in searching, providing information and decision making.

Due to the difficulties of novices to map a problem given to problem solving techniques, an agent based

model is proposed to help them. Instructors are also expected to gain advantage from the proposed model to assist them in teaching. We introduced a novel intelligent agent paradigm based on the Belief, Desire and Intention (BDI)[7]-[9] architecture to perform a mapping of a computational problem to produce problem solving documents. These problem solving documents will be elaborated in further sections.

This paper consists of six sections. The second section explains about the previous works that are related to our research. Then, the proposed agent conceptual model is described in section three. Next, section four, present the experimental results which test the proposed model with three different problems. This is followed by section five the gives a discussion of the proposed model. Lastly, section six concludes the paper and provides briefs suggestions for future works.

2 RELATED RESEARCH

2.1 Visualization Tools without Agents

Conventional computational problem solving techniques consists of flow chart, algorithm/pseudocode, structure chart, Nassi-Shneiderman chart and Warnier-Orr diagram. Researchers have designed visualization tools that integrate problem solving techniques to help novices in program comprehension. Several of them are VIP for C++ [10], Ville [11] and Planani [12] for multiple different languages. These tools are based on source code level that assumed students already have programming knowledge. Another tool which is close to the source code level is RAPTOR [13] that enables students to construct flow charts and the tool will visualize them by

- M.A. Teh Noranis is with the Computer Science Department, Universiti Putra Malaysia, Malaysia, Serdang Selangor 43400. E-mail: nuranis@upm.edu.my
- N. Shahrin Azuan is with the Telekom Research and Development, TM Innovation Centre, Malaysia, Cyberjaya Selangor 63000. E-mail: shahrin@tmrnd.com.my

showing the content of variables and arrays.

A system that adopts kick-start activation was introduced in the beginning of an introductory programming course to provide the deep structure of programming before the surface structure by making the students solve different kinds of problem [14]. Simulation game is another visualization tool that assists students in computational problem solving [15].

R. Kadar and S. Sulaiman proposed Zoom Visual Flow (ZViF) Technique that represents source code in graphical views based on different shapes and colors of Action Icons Control Constructs (AICOS) notations [16].

A 3De synergetic program visualization tool was designed to help novice students to view programming stages starting from designing problem solving, developing code and validating logical flow of the program through visualizing multi-level of program abstraction [17].

2.2 Visualization Tools with Agents

The mentioned visualization tools do not utilize multi agent interactions in their design. Multi agent interactions are software systems that contain more than one agent which communicate with each other. Software agents are reactive systems and able to work in inaccessible, non-deterministic, dynamic and continuous environments compared to functional systems (e.g. a compiler). Therefore, they are more flexible than functional systems where they can make decisions autonomously in the mentioned environments.

Several works support visualization with agents to visualize pseudo-code language [18], visualize expression in C [19] and visualize object parameters [20]. Social agents are used to support the understanding of program visualization through discussion with user and explanation from the agent [18]. [19] adapt the level of details in the visualizations based on the learner's knowledge and focus on topic that are not understood by the user. An approach to change visualization parameters for example shape, colors and style is used to help novices in programming [20]. Jeliot 3 [21] integrates multi agents namely Student Agent, Record Agent, Modelling Agent, Learning Object Agent and Evaluation Agent to provide dynamic and adaptive learning materials to individual users.

Designing multi-agent system requires knowledge on the role of agents. Besides autonomous features, agents are designed to interact with human or systems, synthesize information, monitor, negotiate and coordinate. Agents and Artificial Intelligence (AI) are associated to each other. Several research have integrated agents with fuzzy logic for information extraction [22], controller design [23], healthcare [24], manipulator systems [25], decision making [26].

The mentioned tools differ from the proposed model in terms of the architectural design. To date, no other visualization problem solving tools have modeled agents as what has been proposed. The uniqueness of the design will be explained further in the next section. The feature

of agents as problem solvers and capability to act independently is appropriate to aid novices in computational problem solving.

3 PROPOSED MODEL

The proposed model focuses on novice students that do not have any background in programming.

The scenario is as follows. It starts from a simple problem or exercise given to students in text form. The students are required to understand the problem and transform the problem to problem solving documents consisting of a PAC, HIPO Chart, IPO Chart, Flowchart and Algorithm. These problem solving documents are related to one another. The PAC is based on the problem, the HIPO is based on the PAC, the IPO is based PAC and HIPO, the flowchart and algorithm is based on the IPO. The data that are related will be explained in detail shortly. Therefore, a clear understanding on the problem given is vital. This is to avoid or reduce any programming errors or bugs during program development.

Our proposed model consists of two modules namely Problem Solving Comprehension Module and Assignment Module separated by the dashed line as shown in Figure 1. The agents are modeled using Prometheus notation [7]-[8].

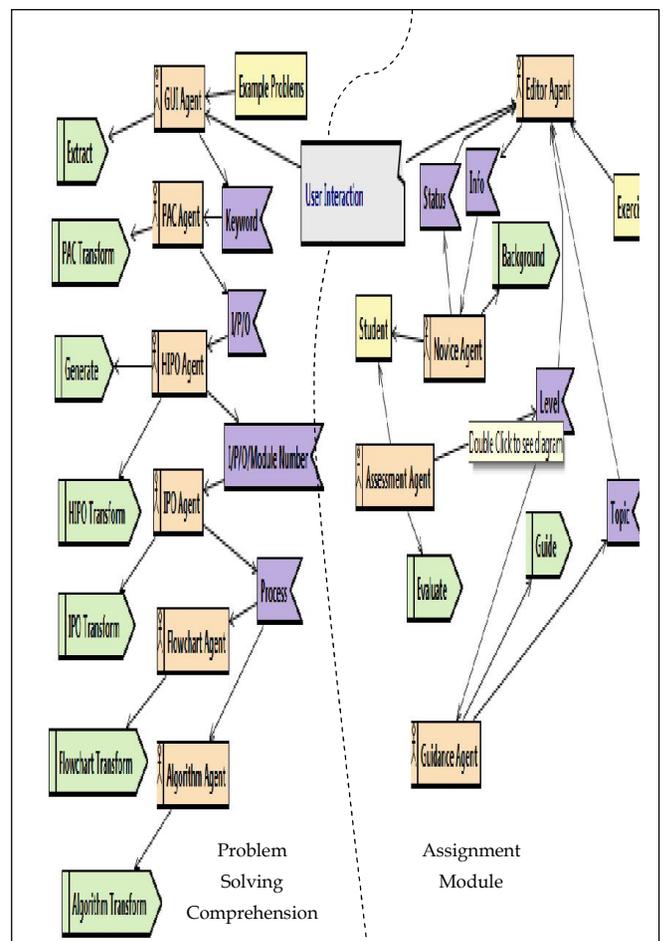


Fig. 1. Proposed Model

3.1 The Problem Solving Comprehension Module

The Problem Solving Comprehension Module provides a database containing various example problems in computer programming based on the most basic course syllabus: fundamental programming, selections, loops, methods, arrays, objects and classes and strings and text input/output. The role of agent in the Problem Solving Comprehension Module is to extract current information starting from the problem given in text form, transform the information to problem solving documents and generate module number. Six agents are needed to perform these tasks namely, GUI Agent, PAC Agent, HIPO Agent, IPO Agent, Flowchart Agent and Algorithm Agent. The extraction algorithm proposed in [27] and fuzzy logic will be integrated in these agent actions. The GUI Agent is the interface that will be used by the user to interact with other agents. These agents communicate with each other by sending data/messages - keywords, input/Process/Output (I/P/O), module number and process.

3.2 The Assignment Module

The Assignment Module provides a database containing exercise to be solved by novices. The exercise covers all the topics mentioned in the Problem Solving Comprehension Module. The roles of agents in the Assignment Module are mainly for interaction with the user starting from obtaining the students background, guiding the students and evaluate the students' level. The agents designed for this module consists of Editor Agent, Guidance Agent, Novice Agent and Assessment Agent. The function of the Editor Agent in this module is the same as the function of GUI Agent in the Problem Solving Comprehension Module. These agents also communicate with each other by sending data/messages - student information, student current status/background, student level/performance and guide students in certain topic. The agents in the proposed system are practical reasoning agents based on the standard BDI architecture. The Jason AgentSpeak interpreter is used to implement the agents [28].

3.3 Class Diagram

Currently, in our research work, we are in the stage of programming the GUI, PAC and HIPO Agent. We designed the Problem Solving Comprehension Module consisting of these agents before coding. Figure 2 shows the class diagrams of these agents. The ProblemSolvingAgentGUI extends the AgArch class and communicates with the guiAgent, pacAgent and hipoAgent. The ProblemSolvingAgentGUI constructor displays the interface containing a button that is used by the novice to start the problem understanding process. The act method executes an action and the stop method is called by the infrastructure tier when the agent is about to be killed.

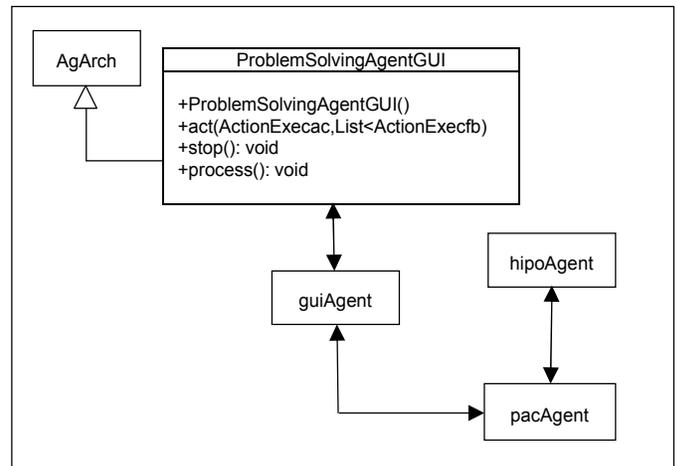


Fig. 2. Agent Class Diagram

3.4 Proposed Extraction Algorithm

The process method in Figure 2 is explained further in Figure 3 as follows, which is the proposed extraction algorithm.

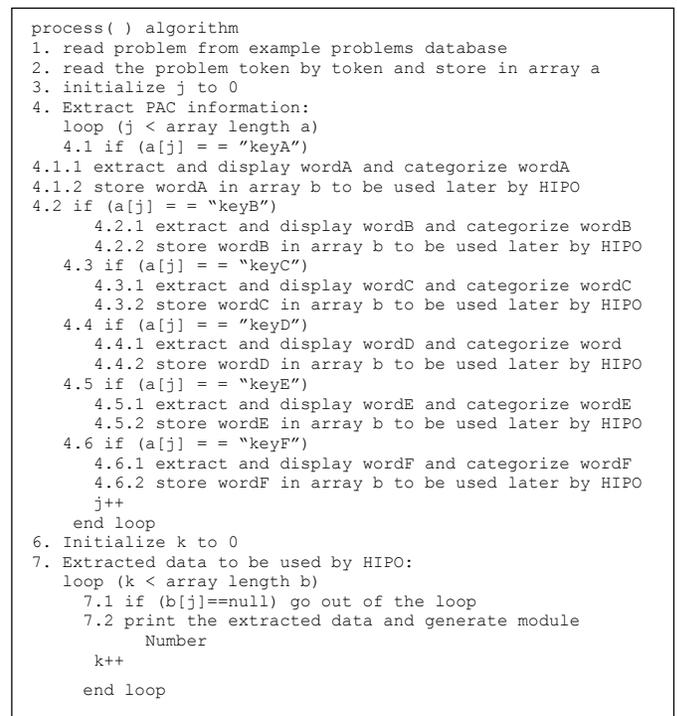


Fig. 3. Proposed Extraction Algorithm

4 EXPERIMENTAL RESULTS

The proposed model is tested with three different problem statements. These problems are written in text form.

4.1 Problem 1

Question: Write a Problem Analysis Chart (PAC) to read test1 and test2 of a student. Find the sum and average using the formulas $sum=test1+test2$ and $average=sum/2$. Display average.

The results are shown below in Figure 4.

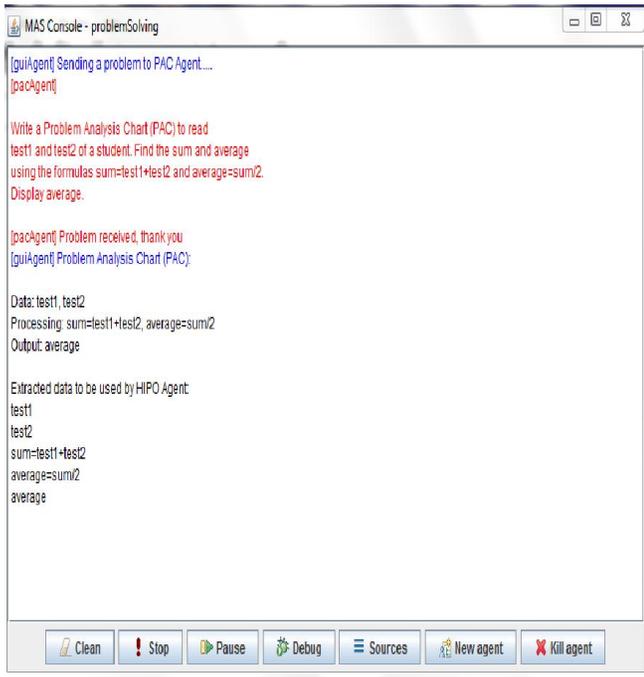


Fig. 4. Problem 1 Screen Shot Result

4.2 Problem 2

Question: Write a PAC to get length and width of a rectangle. Calculate the area of the rectangle using the equation $area=length \times width$. Print area.

The results are shown in Figure 5.

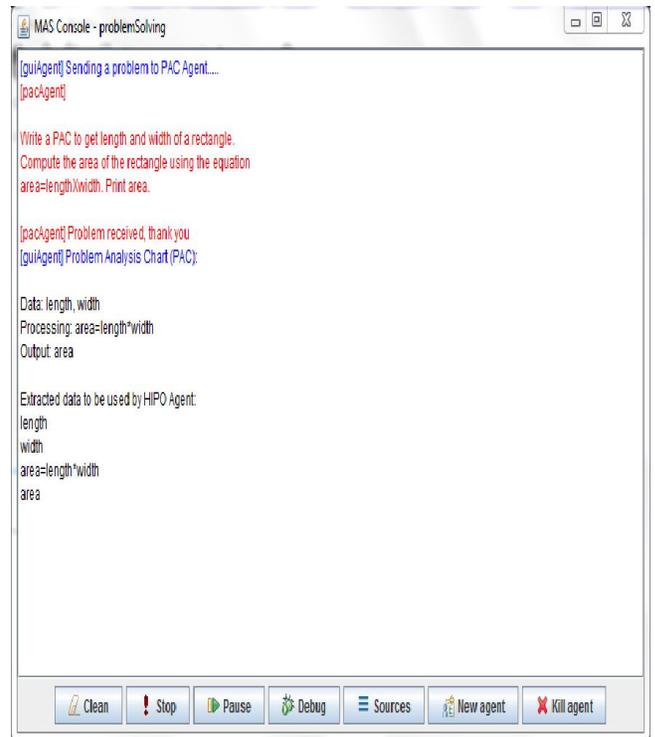


Fig. 5. Problem 2 Screen Shot Result

4.3 Problem 3

Write a PAC to input totalbill and payment by a customer in a supermarket. Calculate $changedue=payment-totalbill$ of the customer. Output totalbill and changedue.

The results are illustrated in Figure 6.

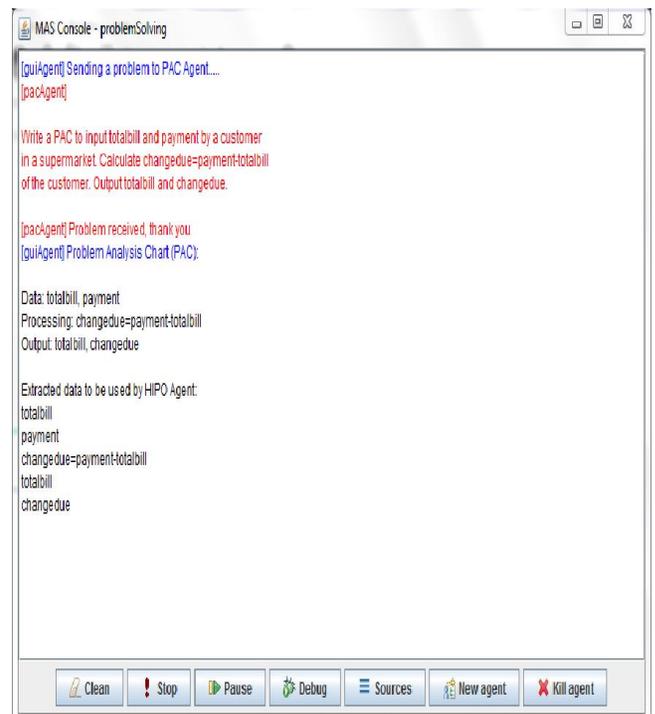


Fig. 6. Problem 2 Screen Shot Result

Figure 7 shows the user interface controlled by the novice to understand the problems that are stored in the Example Problems database.

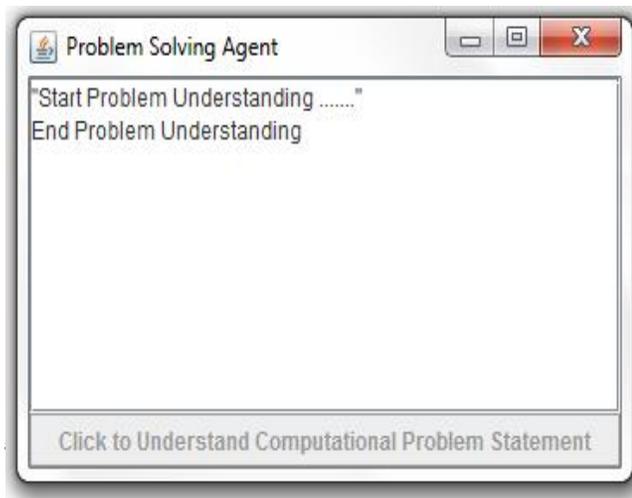


Figure 7. User Interface

5 DISCUSSION

Our proposed model has been tested using the three different problems mentioned and succeeded to produce the accurate results. This is shown by the results in Figures 4, 5 and 6 indicating Data, Processing and Output where the keywords in the problems given in text form are extracted correctly. The results also illustrate communication between guiAgent, pacAgent and hipoAgent that demonstrate knowledge representation by these agents which shows the visualization features.

In our research work, we are using the existing conventional problem solving techniques. Our contribution is mainly on the novel model that is different from other visualization problem solving model. In addition, the extract, transform and generate algorithms in the Problem Solving Comprehension Module; and capture novice background, evaluate and guide in the Assignment Module are also the contributions in this research work. The Problem Solving Comprehension Module function is for students' understanding of a computational problem where a problem is presented and the solution is given. This is shown by the guiAgent, pacAgent and hipoAgent communication. The Assignment Module function is to stimulate students' critical thinking skills through interaction with the user and the agents namely novice agent, editor agent, assessment agent and guidance agent. The editor agent acts as the interface with the novice agent. The assessment agent assesses the level of students' critical thinking ability. We plan to use rubrics as the assessment agent knowledge representation to assess students' critical thinking. The guidance agent role is to guide students to the correct path to solve a particular computational problem.

Furthermore, for the Problem Solving Comprehension Module, we intend to improve our model by adding more keywords to cater various computational problems written in text. We are currently in the process of extending and improving the design and coding of the algorithms mentioned using Jason AgentSpeak [28], an agent programming language which is built on top of the Java programming language. The Assignment Module will be designed in such a way to cater students' critical thinking skills that can be measured by rubrics.

This model represents agents that exhibit the reactive, proactive and social ability features. These features can aid novices that face difficulties in computational problem solving, as problem solving is very important not only in the computer science field but also other fields such as medical, engineering and business. Moreover, this model can enhance the students' computational problem solving skills.

6 CONCLUSION AND FUTURE WORKS

A model based on agents has been created to assist students in computational problem solving. The Problem Solving Comprehension Module function is for extraction, transformation and module number generation. The Assignment Module is for interaction with the user. The agent in the Problem Solving Comprehension Module consist of GUI, PAC, HIPO, IPO flowchart and algorithm agents. The agents in Assignment Module consist of the Editor, Novice, Assessment and Guidance agents. The initial model is evaluated using three different problem statements and produced accurate results. In addition, the proposed model illustrates visualization, communication and knowledge representation features by software agents.

For future works, the design and coding of the whole model that has been built will be implemented. In addition, the proposed extraction algorithm will be extended to provide more keywords to cover a wide range of problem statement which is associated to the data stored in the Example Problems database. Moreover, the proposed problem solving visualization tool will be integrated with a program development tool based on agents.

ACKNOWLEDGMENT

The authors wish to thank Kementerian Pengajian Tinggi for sponsoring this research work under the IPTA Fundamental Research Grant Scheme (FRGS), project code: 02-12-10-1000FR.

REFERENCES

- [1] P. Guo and H. Qi, "Study on Interactive System for Teaching Programming (ISTP) based on Intelligent Agent," *Proc. 2010 6th International Conference on Wireless Communications WiCOM 2010, Networking and Mobile Computing*, pp. 23-25, 2010.
- [2] K.E. Boyer, W. Lahti, R. Philips, M.D. Wallis, M.A. Vouk, and J.C. Lester, "Principles of Asking Effective Questions during Student Problem Solving," *Proc. of the 41 st ACM Technical Symposium on Computer Science Education*, pp. 460-464, 2010.
- [3] W. Pedrycz and P. Rai, "A Multifaceted Perspective at Data Analysis: a Study in Collaborative Intelligent Agents," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 38, no. 4, pp. 1062-1072, 2008.
- [4] S. Yaskawa and A. Sakata, "The Application of Intelligent Agent Technology to Simulation," *Mathematical and Computer Modeling*, vol. 37, no. 9-10, pp. 1083-1092, 2003.
- [5] B.L. Iantovics, "Agent-based Medical Diagnosis Systems," *Computing and Informatics*, vol. 27, no. 2, pp. 593-625, 2008.
- [6] D.A. Bell, A. Beck, P. Miller, Q.X. Wu, and A. Herrera, "Video Mining - Learning Patterns of Behaviour via an Intelligent Image Analysis System," *Proc. of the 7 th International Conference on Intelligent Systems Design and Applications*, pp. 460-464, 2007.
- [7] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*. Chichester: Wiley, 2004.
- [8] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley United Kingdom, 2009.
- [9] Wikipedia Foundation, Inc. http://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model
- [10] A.T. Virtanen, E. Lahtinen, and H.-M. Järvinen, "VIP, A Visual Interpreter for Learning Introductory Programming with C++," *Proc. of the Fifth Finnish/Baltic Sea Conference on Computer Science Education*, pp. 129-134, 2005.
- [11] T. Rajala, M.-J. Laakso, E. Kaila, and T. Salakoski, "VILLE - A Language-Independent Program Visualization Tool," *Proc. of the Seventh Koli Calling Conference on Computer Science Education*, 2007.
- [12] J. Sajaniemi and M. Kuitinen, "Visualizing Roles of Variables in Program Animation," *Information Visualization 3*, pp. 137-153, 2004.
- [13] J.C. Giordano and M. Carlisle, "Toward a More Effective Visualization Tool to Teach Novice Programmers," *Proc. of the 7 th Conference on Information Technology Education*, pp. 115-122, 2006.
- [14] E. Lahtinen and T. Ahoniemi, "Kick-Start Activation to Novice Programming - A Visualization-Based Approach," *Electronic Notes in Theoretical Computer Science*, vol. 224, pp. 125-132, 2009.
- [15] C.C. Liu, Y.B. Cheng, and C.W. Huang, "The Effect of Simulation Games on the Learning of Computation Problem Solving," *Computers & Education*, vol. 57, pp. 1907-1918, 2011.
- [16] R. Kadar, and S. Sulaiman, "The Effectiveness of Zoom Visual Flow (ZViF) Technique in Program Comprehension Activities," *International Symposium in Information Technology (ITSim)*, pp. 1-6, 2010.
- [17] Affandy, N. Suryana, S. Salam, M.S. Azmi, and E. Noersasongko, "3De - Synergetic Program Visualization: A Visual Learning Tool for Novice Students," *International Conference on E-Education, Entertainment and E-Management (ICEEE)*, pp. 133-137, 2011.
- [18] R. Miraftebi, "Intelligent Agents in Program Visualizations: A Case Study with Seal," *Proc. of the First International Program Visualization Workshop*, pp. 53-58, 2011.
- [19] P. Brusilovsky, "Adaptive Visualization Component of a Distributed Web-Based Adaptive Educational System," *Intelligent Tutoring System*, Cerri, S.A., Gourderes, G., Paraguoou eds., LNCS, Springer-Verlag, vol. 2363, pp. 229-238, 2002.
- [20] M. Lattu, J. Tarhio, and V. Meisalo, "How a Visualization Tool Can be Used - Evaluating a Tool in A Research & Development Project," *Proc. of the 12th Psychology of Programming Interest Group (PPIG) Workshop*, 2000.
- [21] M. Ben-Ari, R. Bednarik, R.B-B. Levy, G. Ebel, A. Moreno, N. Myller, and E. Sutinen, "A Decade of Research and Development on Program Animation: The Jeliot Experience," *Journal of Visual Languages and Computing*, vol. 22, pp. 375-384, 2011.
- [22] J. Ropero, A. Gómez, A. Carrasco, and C. León, "A Fuzzy Logic Intelligent Agent for Information Extraction: Introducing a New Fuzzy Logic-based Term Weighting Scheme," *Expert Systems with Applications*, vol. 39, pp. 4567-4581, 2012.
- [23] E.A. Olajubu, O.A. Ajayi, and G.A. Aderounmu, "A Fuzzy Logic Based Multi-Agents Controller," *Expert Systems with Applications*, vol. 38, pp. 4860-4865, 2011.
- [24] G. Fenza, D. Furno, and V. Loia, "Hybrid Approach for Context-aware Service Discovery in Healthcare Domain," *Journal of Computer and System Sciences*, vol. 78, pp. 1232-1247, 2012.
- [25] P. Bohner, "A Multi-Agent Approach with Distributed Fuzzy Logic Control," *Computers in Industry*, vol. 26, pp. 219-227, 1995.
- [26] S. Li and J. Zheng Li, "AgentsInternational: Integration of multiple agents, simulation, knowledge bases and fuzzy logic for international marketing decision making," *Expert Systems with Applications*, vol. 37, pp. 2580-2587, 2010.
- [27] T.N. Mohd Aris, "Object-Oriented Programming Semantics Representation Utilizing Agents," *Journal of Theoretical and Applied Information Technology*, vol. 31, pp. 10-20, 2011.
- [28] M. Gustave, "Jason A Java-based Interpreter for an Extended Version of AgentSpeak," <http://jason.sourceforge.net/Jason/Jason.html>. 1865.

M.A. The Noranis received her BSc in Information Technology from University Utara Malaysia in 1994 and MSc in Artificial Intelligence from University Putra Malaysia in 2002. She received her PhD in Computer Science from University Kebangsaan Malaysia in 2008. She had been working as a programmer and system analyst before she joined University Putra Malaysia as a tutor. She is currently a Senior Lecturer at the Department of Computer Science, Faculty of Computer Science and Information Technology, University Putra Malaysia. Her research interests include Artificial Intelligence, parallel processing, programming science, software agents and bio-nanorobotics.



N. Shahrin Azuan obtained his bachelor degree in Computer Science from Clarkson University, NY, U.S.A. followed by master degree of Information Systems Engineering from UMIST, Manchester, U.K. and PhD degree in Electrical Engineering from University Teknologi Malaysia. He is currently a Principal Researcher at Telekom Malaysia Research and Development. His research interests include computational intelligence, biometric technology, information security and networking.