Proc. of the Second Intl. Conf. on Advances in Computer and Information Technology - ACIT 2013

ISBN:978-981-07-6261-2

A Multi-Agent System for Computational Problem Solving -A Review

Maryam Rajabi
Department of Computer Science
Universiti Putra Malaysia 43400
Selangor, Malaysia

Teh Noranis Mohd Aris
Department of Computer Science
Universiti Putra Malaysia 43400
Selangor, Malaysia

***Abstract*— Computational problem solving is taken into account as the first step before source code development. However, novice students face difficulties in understanding problem statements and transforming them to computational problem solving techniques. This requires the understanding of fundamental programming concepts. It is very important to cater this problem from the beginning before writing the source codes. This article reviews current studies on the designed environments for problem solving and the possibility to propose a new architecture for computational problem solving utilizing multi-agent technology.**

***Keywords*— Multi-Agent, Computational Problem Solving, Computer Programming.**

I. INTRODUCTION

Today, computer programming is known as a core basic skill, not only essential for those students studying in computer science and software engineering, but also required in other technology disciplines including physical sciences as well as engineering [1]. Programming problems are rooted from a huge number of problem domains. Comprehending the problem domain is of significance. In addition, novices seem to suffer from the incapability to combine the individual statements and constructs related to a programming language into valid programs [2]. Multi-agent systems are defined as a set of, heterogeneous, computational entities which have their own specific problem solving capabilities and can make some interaction among them aimed at achieving an overall goal [3] [4]. Regarding capability of multi agent systems, it should be noted that multi-agent systems usually have been utilized in industrial applications but merely has been used to help in designing of programming environments [5] -[10].

In this study, our main focus of study is to investigate the possibility to use multi-agent systems as a new agent model for computational problem solving to help the novices.

II. RELATED WORK

A. Computer programming

Programming is considered as a 'mental tool' of general interest, a complex task which consists of comprehending, method finding and also coding [1]. Programming can be divided into four basic steps as follows : a) Comprehending the problem at hand b) Defining a solution to the existing problem, at the outset in any form c) Translating that form into a selected programming language and d) Testing and debugging the resulting program [2]. Students are seldom seemed to be aware of the problems which are more likely to be solved by a computer and the benefits related to using programming, so they face serious difficulties. On the other hand, programming is supported by means of some professional integrated software environments; which seemed useless for novices [1] [2] [11]. Some frustration feeling among the students working with programming often is associated with the programming environment in use [12]. Well-known environments that currently have been designed for learning programming in computer language C, data mining and other languages consist of [1] [2] [12] -[14]:

1- L.E.C.G.O. (Learning Environment for programming and C using Geometrical Objects)

2- PASS (Programming Assignment assessment System)

3- Object-Karel

4- WYSIWYC (What You See Is What You Code)

5- ONTOIAS (ontology-supported information agent shell)

B. L.E.C.G.O

L.E.C.G.O. is viewed as an open problem-solving computer learning environment designed in order to make supports for secondary level education students and first year university students when learning programming and C [2]. As "Fig. 1**"**, shows that the design of L.E.C.G.O. was resulted from synthesizing and combination of three following models: a) the learning model b) the subject matter model, and c) the learner model. The two following main parts make the general architecture of L.E.C.G.O. : a) Offering the suitable content for learning essentials in programming and C, and b) dedicating the learning activities which students have to follow and perform [2] [12] [13]. Assistance is also provided in four modes: i) as ready specific expressions ii) ready structures and functions in pseudo-code iii) ready structures and functions in C iv) as appropriately designed content. In order to evaluate L.E.C.G.O, we considered a pilot study of the context which is designed in terms of the following features:

Proc. of the Second Intl. Conf. on Advances in Computer and Information Technology - ACIT 2013

ISBN:978-981-07-6261-2

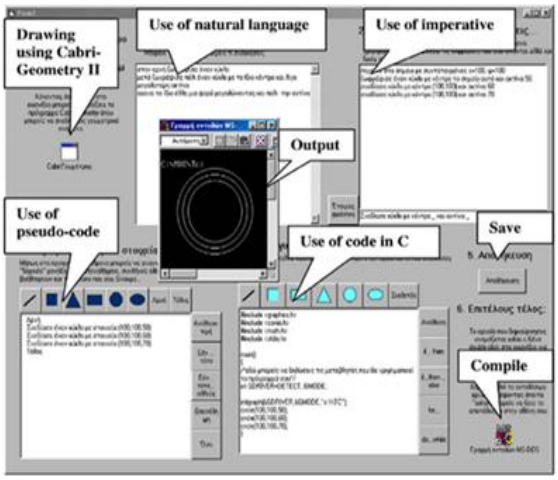

Figure1. The main window of the programming environment [14].

1- Qualitative study
2- Nine students, 18 year-old, 2hours/task
3- Three learning settings including paper–pencil, Turbo C and L.E.C.G.O.
4- Four learning tasks/setting
5- Data: students' programs and writings in all Representation System(RS), the researcher's notes [2].

The finding of the study shows that L.E.C.G.O also could be easily modified for learning any programming language.

C. PASS

PASS is defined as a program submission/assessment system firstly developed in 2004 in City University of Hong Kong, with the main objective of assisting and contributing the starters in learning programming like data mining courses as well as data structures [1]. PASS is typically used following teaching some specific set of programming skills; the instructor uploads a programming problem, along with a couple of test cases [15]. The associations among the core entities in PASS are summarized in Figure "Fig. 2" [16]. The results obtained show that it can assist the novices to check their lab exercises by themselves and carry out all their lab exercises. The novices are also able to work more in an independent manner and it makes them feel more confident when they got all correct. Gradually, they will be more independent. On the other hand, novices can identify the bugs immediately; it enhances their learning rate. Regarding above obtained results, a well facilitated e-learning setting can increase their learning motivation and self-efficacy [1] [15] [16].

D. Object-Karel

Object-Karel is a programming environment to help the students to develop programs in an easy manner. Regarding "Fig. 3", Object-Karel incorporates e-lessons, including both theory and hands on activities which help in the teaching programming. Using this environment, the beginners will become familiar with the specific unit concepts instead of writing a program from the beginning and they are granted the opportunity to do experiments by ready examples.

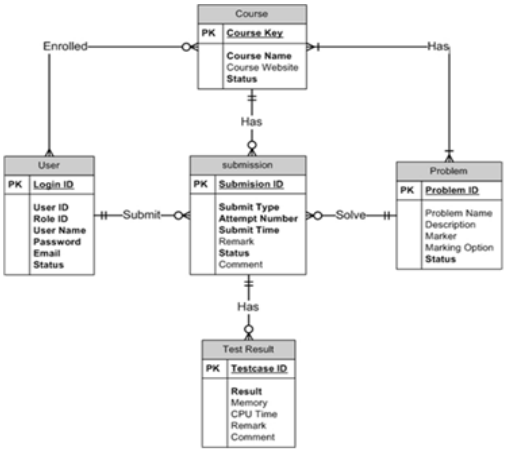

Figure 2. Relationships among core entities in pass [16].

The achieved result shows that 79% of students reckon that the e-lessons with theory and activities significantly helped both teaching and learning process while 21% answered that they did not know if this contributes or not. All students also mentioned that the structure editor assisted them to develop their program. Likewise, the average score for the eight criteria of usability is 4.4 out of 5 [14].

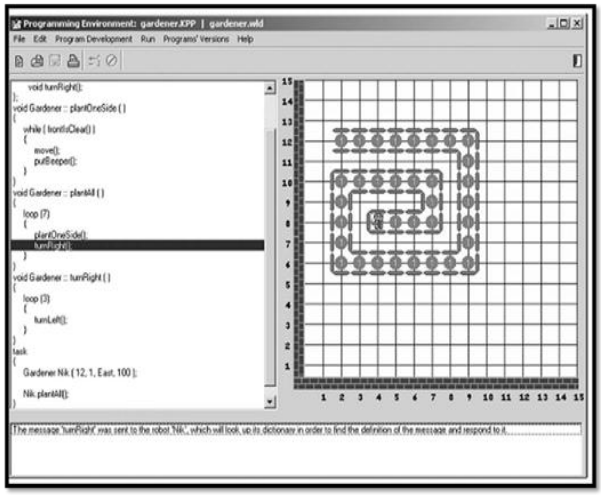

Figure 3. The main window of the programming environment [14].

E. WYSIWYC

WYSIWYC as a ''live'' development model can narrow the gap between coding an algorithm and that viewing its visualization. It also acts as an alternative editing model that provides, on an edit-by-edit basis, immediate visual feedback on the syntactic and semantic correctness of code. In the WYSIWYC model, a native learner develops an algorithm through a combination of typing into a program editor and directly manipulating program objects. In fact the advance learners can construct their own visualizations of algorithms in order to foster active learning. Researchers have implemented this model in a new version of the ALVIS software [11] called ALVIS LIVE!. Within an introductory computer science

Proc. of the Second Intl. Conf. on Advances in Computer and Information Technology - ACIT 2013

ISBN:978-981-07-6261-2

course, ALVIS LIVE! enable students to develop semantically correct algorithms with minimal intervention from a teaching assistant. In other words, ALVIS LIVE! apply to develop semantically correct algorithms with minimal intervention from a teaching assistant. In the environment WYSIWYC model, SALSA (Simple Actor Language System and Architecture) pseudocode language is an actor-oriented programming language that uses concurrency primitives beyond asynchronous message passing, including token-passing, join, and first-class continuations. It also supports distributed computing over the Internet with universal naming, remote communication, and migration linguistic abstractions and associated middleware. For portability, it produces Java code.

"Fig. 4" show annotated snapshot of the ALVIS LIVE! environment. SALSA pseudocode can be directly typed into the script window. Alternatively, toolbox tools can be used within the animation window to generate many commands by direct manipulation.

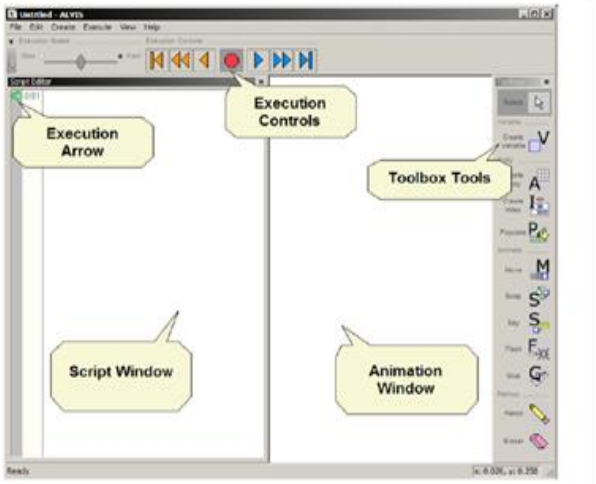

Figure 4. Annotated snapshot of the ALVIS LIVE! environment [11].

The results obtained from the WYSIWYC development model showed great promise in helping participants both to identify and correct program errors quickly, and ultimately to understand the execution of the code they were writing.

III. AGENTS AND MULTI AGENTS

An agent is anything viewed as perceiving its environment by means of sensors and actuators based on the environment through effectors [2]. Group agents are divided into the following five classes in terms of their perceived intelligence and capability degree [17]:

1- Simple reflex agents

2- Model-based reflex agents

3- Goal-based agents

4- Utility-based agents

5- Learning agents

Recently, Multi-Agent System (MAS) is viewed as a system included with multiple interacting intelligent agent proposed for modelling complex systems and solving large scale problems as has been shown in "Fig. 5" [18]. Multi-agent systems can also be used in order to solve the difficult or impossible problems for an individual agent or a monolithic system. In addition to supporting e-learning, multi-agent systems (MAS) can support high-ranking demand like solving problems in an automatic manner [17] [18].

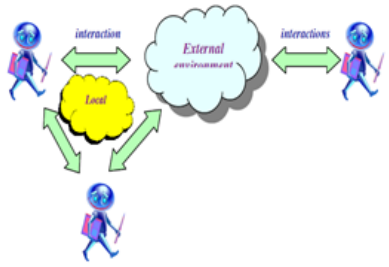

Figure 5. Multi agent system [17].

A. ONTOIAS

As has been mentioned in last section multi-agent systems are able to be used in programming environments. An OntoIAS (ontology-supported information agent shell) as an environment which was being applied multi agent technique [4]. It is the key section of the information agent detached from the domain ontology with the extending application of information area of agent shell based on the ontology itself, making it possible for the semantic information bases to share and communicate to automatically assimilate and integrate ontologies associated. The OntoIAS architecture diagram consists of the four main modules of information agents, such as information searching, information extracting, information classifying, and information presenting/ranking and corresponding to OntoCrawler, OntoExtractor, OntoClassifier, and OntoRecommander, which respectively have been designed as in"Fig. 6". In other words, not only can OntoIAS quickly search, extract, classify, and integrate specific domain documents, but it can also precisely recommend important information to allow excellent information integration and recommendation ranking [4].

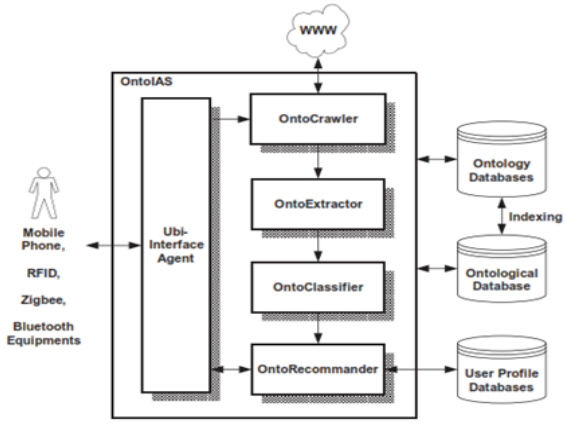

Figure 6. Conceptualized architecture of ontoias [4].

Ontology is a theoretical philosophy which principally explores the knowledge characteristics of life and real objects;

Proc. of the Second Intl. Conf. on Advances in Computer and Information Technology - ACIT 2013

ISBN:978-981-07-6261-2

Table 1: A summary of investigating environments

Tool Name	Feature	Source Code/Problem Solving	Technique	Experiment/performance	Language	Case Study	Year
Pass	A program submission/assessment system	Pseudocode	Non- Agent	To enhance learning motivation; efficacy, e-effect	C	City university of Hong Kong, data structures and data mining courses.	2004
L.E.C.G.O	An open problem-solving computer learning environment	Algorithm	Non- Agent	Feasibility and speed of programing	C	Nine students-12th grade (18year-old, 2hours/task), secondary school in Patras, Greece.	2006
ONTOIAS	An ontology-supported information agent shell for ubiquitous services	Flow chart	Agent	Reliability, validity, speed of system processing time; to predict user information requirements for reducing system processing and users waiting time	JAVA	Taiwan universities	2008
Object-Karel	A designed environment for the teaching of OOP for novice programmers	Pseudocode	Non- Agent	Learning and teaching motivation	Karel++	19 students from the department of applied informatics of the university of acedonia, Greece.	2002
WYSIWYC	A ‘‘live’’ algorithm development and visualization model for introductory computer science students	Algorithm	Non- Agent	To develop semantically correct code, To immediate syntactic feedback	SALSA	A computer science course at Washington state university consist of 80 students in four laboratory sections .	2004

ISBN:978-981-07-6261-2

in the artificial intelligence field. They used to make definition of domain knowledge content, express knowledge, as well as solving communication and other commonly shared problems. Ontology offers complete semantic models, meaning in which all related entities, characteristics and knowledge base become in specified domains. As a result, ontology is a powerful tool for constructing and maintaining information systems. Table 1 are the summarized all the programming environments that are investigated in this study.

IV. DISCUSSION

In this work, based on reviewing literature on computer programming, a good performance in programming is able to make contributions to learners to make use of various and new representation systems in order to express their problem solving strategies. Here, L.E.C.G.O as a Learning Environment can be easily modified for any programming language learning and PASS helps novices to easily check their own lab exercises and feel more confident and Object-Karel is a programming environment to help the students to develop programs in an easy manner. In addition, the WYSIWYC development model showed great promise in helping participants both to identify and correct program errors quickly, and ultimately to understand the execution of the code they were writing and also OntoIAS quickly search, extract, classify, and integrate specific domain documents. Pedagogical Algorithm Visualization (AV) systems produces graphical representations that aim to assist learners in understanding the dynamic behavior of computer algorithms. In the other hands, technology graphically illustrates how algorithms work conventional computational problem solving techniques range from the flow chart, algorithm/pseudocode to structure chart. These problem solving techniques is characterized in the form of visualization tools, which can be regarded as without using agent technology and using agent technology. In addition, a model based on agents can contribute students in computational problem solving. Associated with complex problems, multi-agent systems can be taken into account as a perfect agent to solve it since multi-agent systems have the features of flexibility and intelligence, solving complex issues in terms of distributed knowledge with updating capabilities. Finally, results of this study also showed that a well facilitated e-learning setting is able to increase learning motivation and self-efficacy. Our contribution in this paper is to investigate the possibility of a new environment to assist novice programmers in computational problem solving using agents.

V. CONCLUSION AND FUTURE WORK

In this paper, we have compared several tools and we want to come out with a new tool for computational problem solving applying multi-agent because multi-agent systems exhibit flexibility and intelligence to solve complex issues in terms of distributed knowledge with updating capabilities.

For future works, we will design the problem solving model with our own architecture from different other visualization model. The design and coding of the whole proposed model will be implemented.

ACKNOWLEDGMENT

I would like to thank my supervisor Dr. Teh Noranis Mohd Aris for her guidance. I would also like to thank Kementerian Pengajian Tinggi (KPT) for the support given under the Fundamental Research Grant Scheme (FRGS) University Putra Malaysia, project code number 02-12-10-1000FR.

REFERENCES

[1] Kris, M.Y. L., Victor, C.S. L. & Yu, T.Y. (2010). Learning motivation in e-learning facilitated computer programming courses. Computers & Education 55: 218–228.

[2] Kordaki, M. A. (2010). drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. Computers & Education 54(1): 69-87.

[3] Nguyen T. M. K.,(2012). Developing an Intelligent Multi-Agent System based on JADE to solve problems automatically.International Conference on Systems and Informatics (ICSAI 2012).

[4] Yang, S.Y., (2011). OntoIAS: An ontology-supported information agent shell for ubiquitous services. Expert Systems with Applications38: 7803–7816.

[5] Pakonen, A., Pirttioja, T., Selionen, I., & Tommila, T. (2006). Proactive computing in process monitoring: Information agents for operator support. In Proceedings of IEEE conference on emerging technologies and factory automation: 153–158.

[6] Pakonen, A., Tommila, T., Pirttioja, T., & Selionen, I. (2007). OWL based information agent services for process monitoring. Conference, Patras, Greece: 9–16.

[7] Pirttioja, T., Pakonen, A., Seilonen, I., Halme, A., & Koskinen, K. (2005). Multi-agent based information access services for condition monitoring in process automation. Conference, Perth, Australia:240-245.

[8] Gao, S., Wang, H., Wang, Y., Shen, W., & Yeung, S. (2005). Web service-agents-based family wealth management system. Expert Systems with Applications, 29: 219–228.

[9] Wu, S. H. (2007). Agent-mediated ebXML application system. Master thesis, Dept. of Computer Science and Information Engineering, TaTung University, Taipei, Taiwan.

[10] Belmonte, M. V., Pe´rez-de-la-Cruz, J. L., & Triguero, F. (2008). Ontologies and agents for a bus fleet management system. Expert Systems with Applications 34: 1351–1365.

[11] Hundhausen, C.D., & Brown J.L.,(2007).What You See Is What You Code: A "live" algorithm development and visualization environment for novic learners. Journal of Visual Languages & Computing18: 22–47.

[12] Kordaki, M. (2009). Beginners' programming attempts to accomplish a multiple-solution based task within a multiple representational computer environment. Proceedings of World Conference on Educational Multimedia: 3282-3295.

[13] Kordaki, M. (2006). Learning activity' as the basic structural element for the design of web-based content: A case study. Proceedings of World Conference on E-Learning in Corporate. Government, Healthcare, and Higher Education: 88-96.

[14] Stelios X., Maya S. and Vassilios D., (2006). An introduction to object oriented programming with a didactic microworld: objectKarel. Computers & Education47:148–171.

[15] Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. Computer Science Education 15: 83–102.

[16] M. Choy, S. Lam, C.K. Poon, F.L. Wang, Y.T. Yu, & L. Yuen., (2008). Design and implementation of an automated system for assessment of computer programming assignments: Advances in web-based learning. Conference, Berlin/Heidelberg: 584–596.

[17] Zhuo, L., (2012). A Multi-Agent Based Approach for Solving the Redundancy Allocation Problem. Proquest, Umi Dissertation Publishing.

[18] Gehao, L.u., Joan, L.u., Shaowen, Y.a.o., & Jim, Y.i.p., (2009). A Review on Computational Trust Models for Multi-agent Systems. The Open Information Science Journal 2: 18-25.