UNIVERSITI PUTRA MALAYSIA

AGENT-BASED EXTRACTION ALGORITHM FOR COMPUTATIONAL PROBLEM SOLVING

MARYAM RAJABI

FSKTM 2015 6

AGENT-BASED EXTRACTION ALGORITHM FOR COMPUTATIONAL PROBLEM SOLVING

By

MARYAM RAJABI

© COPYRIGHT UPM

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfillment of the Requirement for the Degree of Master of Science

October 2015

All material contained within the thesis, including without limitation text, logos, icons, photographs, and all other artwork, is copyright material of Universiti Putra Malaysia unless otherwise stated. Use may be made of any material contained within the thesis for non-commercial purposes from the copyright holder. Commercial use of material may only be made with the express, prior, written permission of Universiti Putra Malaysia.

Copyright© Universiti Putra Malaysia

© COPYRIGHT UPM

DEDICATION

Dedicated to My Father, Mother
and My Brothers

© COPYRIGHT UPM

Abstract of thesis presented to the Senate of Universiti Putra Malaysia
in fulfilment of the requirement for the degree of Master of Science

AGENT-BASED EXTRACTION ALGORITHM FOR COMPUTATIONAL PROBLEM SOLVING

By

MARYAM RAJABI

October 2015

Chair: Teh Noranis Binti Mohd Aris, PhD
Faculty: Computer Science and Information Technology

Modeling Computational Problem Solving (CPS) is the main issue in teaching programming where the given text is needed to be transformed into a model then later into a programming language. Various CPS techniques have been proposed such as PAC, IPO, Flowchart and Algorithm. Also, various models have been proposed for these techniques either using agent or non-agent based solutions, but the literature shows the existing models and techniques still have limitation for novice programmers as they need prior knowledge on programming language before being able to do CPS. On the other hand, agent-based model offers a good method to solve complex computational problems. Therefore, utilizing the agent-based featured agents for this purpose is a benefit with the aim of helping novice programmers to understand the CPS without knowing the programming language. Therefore, this research aims at developing a tool for this purpose which can be classified into two objectives. The first is to propose an agent based model for CPS and the second is to evaluate the effectiveness of the prototype based on three factors consisting of understanding, efficiency and usability. Four agents have been proposed as an agent based model for CPS, which are *User_Agent*, *PAC_Agent*, *IPO_Agent* and *Algorithm_Agent*. *User_Agent* is responsible for receiving the problem from user and sending it to *PAC_ Agent*. Moreover, the extraction algorithm is located in *User_Agent* to perform extraction of appropriate information needed from the text and later send this information to *PAC_Agent* for analysis modeling and displaying input, process and output. Additionally, *IPO_Agent* not only produces the same PAC's output results, but also generates module number and represents these results in another window. Finally, *Algorithm_Agent* employs the extracted information provided by *IPO_Agent* to produce the pseudo-code of the given problem and shows it in separate window. The proposed agent-based model for CPS has been designed using Prometheus Design Tools (PDT), which can be plugged-in with *Eclipse* Environment. The agent-based model for CPS has been developed in *JADE* environment which applied the basic *Believe, Desire and Intention (BDI)* architecture to support multi-agent system environment. The performance of the tool has been tested using 20 data sets of scenario

© COPYRIGHT UPM

CPS, created by the researcher. The tool is used to evaluate the accuracy of the proposed extraction algorithm to extract the appropriate information which are needed. The results show that the extraction algorithm has been able to extract 100 % of the information correctly. How far the proposed agent-based model for CPS is able to help novice programmer is evaluated by conducting a group based experiment with 35 students from Faculty of Computer Science and Information Technology (FSKTM). Several statistical tests such as normality test and reliability analysis were conducted. The basic statistical frequency method was used to analyse the performance result. The results indicate that the students' understanding of the CPS techniques are improved by using the proposed tool. From the result, the proposed tool has obtained rank 4.17 for "understanding", 4.01 for "efficiency" and 4.07 for "usability". With such results, it can be concluded that our proposed agent-based tool is able to help novice programmer in CPS.

© COPYRIGHT UPM

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk Iijazah Master Sains

ALGORITMA PENGEKSTRAKAN BERASASKAN AGEN UNTUK PENYELESAIAN MASALAH PENGKOMPUTERAN

Oleh

MARYAM RAJABI

Oktober 2015

Pengerusi: Teh Noranis Binti Mohd Aris, PhD
Fakulti: Sains Komputer dan Technologi Maklumat

Model Penyelesaian Masalah Komputeran (CPS) adalah isu utama di dalam pengajaran pengaturcaraam di mana teks yang diberikan perlu diubah menjadi model dan kemudian kepada bahasa pegaturcaraan. Pelbagai teknik CPS telah dicadangkan seperti PAC, IPO, CartaAlir dan Algoritma, juga pelbagai model telah dicadangkan untuk teknik-teknik tersebut samada menggunakan agen atau penyelesaian tanpa agen, tetapi tinjauan literatur menunjukkan model dan teknik yang sedia ada masih mempunyai kekangan untuk pengaturcara baru kerana mereka memerlukan pengetahuan di dalam bahasa pegaturcaraan terlebih dahulu sebelum dapat melakukan CPS. Sebaliknya, model berasaskan agen menawarkan satu kaedah yang baik untuk menyelesaikan masalah pengkomputeran yang kompleks. Oleh itu, menggunakan ciri-ciri berasaskan agen untuk tujuan ini adalah lebih bermanfaat dengan tujuan membantu pengaturcara baru untuk memahami CPS tanpa mengetahui bahasa pengaturcaraan. Maka, tujuan kajian ini adalah untuk membangunkan satu peralatan untuk tujuan tersebut yang boleh dikelaskan kepada dua objektif. Objektif pertama ialah mencadangkan satu model berasaskan agen untuk CPS dan yang kedua ialah untuk menilai keberkesanan prototaip berdasarkan tiga faktor yang terdiri daripada kefahaman, kecekapan dan kebolehgunaan. Empat agen telah dicadangkan sebagai model berasaskan agen untuk CPS, iaitu *User_agent, PAC_Agent, IPO_Agent* dan *Algorithm_Agent*. *User_Agent* bertanggungjawab untuk menerima masalah daripada pengguna dan menghantar masalah tersebut kepada *PAC_Agent*. Selain itu, algoritma pengekstrakan terletak di User_Agent untuk melaksanakan pengekstrakan untuk maklumat yang sesuai yang diperlukan daripada teks dan kemudian maklumat tersebut dihantar kepada *PAC_Agent* untuk analisis pemodelan dan memaparkan input, proses dan output. Selain itu, *IPO_Agent* bukan sahaja memberikan hasil keluaran PAC yang sama, tetapi juga menjana nombor modul dan memaparkan keputusan di tetingkap lain. Akhir sekali, *Algorithm_Agent* menggunakan maklumat yang diberikan oleh *IPO_Agent* untuk menghasilkan Pseudokod bagi masalah yang diberikan dan memaparkannya di tetingkap yang berlainan. Model berasaskan agen untuk CPS yang dicadangkan telah direkabentuk menggunakan *Prometheus Design Tools (PDT),* yang boleh disambungkan dengan persekitaran *Eclipse*. Model berasaskan agen untuk CPS

© COPYRIGHT UPM

telah dibangunkan di dalam persekitaran JADE yang telah mengaplikasiakan senibina basic *Believe, Desire dan Intention (BDI)* untuk menyokong persekitaran sistem agen pelbagai. Prestasi tool ini telah diuji menggunakan 20 set data mengenai senario CPS, yang telah dibina oleh penyelidik. Tool ini digunakan untuk menilai ketepatan algoritma pengekstrakan yang dicadangkan untuk mengekstrak maklumat bersesuaian yang diperlukan. Hasil dapatan menunjukkan bahawan algoritma pengekstrakan ini mampu mengekstrak 100% maklumat secara tepat. Sejauh mana model berasaskan agen untuk CPS yang dicadangkan ini mampu untuk menolong pengaturcara baru telah dinilai dengan menjalankan eksperimen berasaskan kumpulan dengan 35 orang pelajar daripada Fakulti Sains Komputer dan Teknologi Maklumat (FSKTM). Beberapa ujian statistik seperti ujian normal dan analisis kebolehpercayaan telah dijalankan. Kaedah frekuensi asas statistik digunakan untuk menganalisis hasil prestasi. Hasil dapatan menunjukkan bahawa pemahaman pelajar berkenaan teknik CPS adalah meningkat dengan menggunakan peralatan yang dicadangkan. Daripada hasil dapatan, peralatan yang dicadangkan telah mendapat pangkat 4.17 untuk "kefahaman", 4.01 untuk "kecekapan" dan 4.07 untuk "kegunaan". Dengan keputusan ini, ia dapat disimpulkan bahawa peralatan berasaskan agen yang telah dicadangkan dapat membantu pengaturcara baru dalam CPS.

© COPYRIGHT UPM

ACKNOWLEDGEMENTS

First and foremost, I thank God who gave me a chance to be a student in Malaysia which I shall always have wonderful memories of. There are many people I would like to express my appreciation to for helping me to complete this project. My sincere thanks and gratitude goes to my dear advisor Dr. Teh Noranis Mohd Aris for her continuous support in my Master. I am indeed very fortunate to have had such a good advisor and mentor for her inspiration and constructive feedback from the start. Also, I would like to thank my dear co-supervisor Assoc. Prof. Md. Nasir Sulaiman for his kindness, and valuable comments that helped improve this manuscript. He motivated me to go on whenever I felt discouragement. Also, I would like to express my thanks to the academic staffs and administrator staffs of Computer Science and Information Technology faculty of UPM University. 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.
I also owe significant debt of gratitude to my brother, Dr. Javad Rajabi and my cousins, Dr. Mojtaba Rajabi and Dr. Khadijeh Aghaei who never left me homesick. My sincere gratitude extends to my family, to my dear grandmother, my parents, my brothers for the love and constant support during my studies. Last but not least, I would like to acknowledge everyone who has helped me in one way or another, without which this would not have been possible.

© COPYRIGHT UPM

I certify that a Thesis Examination Committee has met on 21 October 2015 to conduct the final examination of Maryam Rajabi on her thesis entitled "Agent-Based Extraction Algorithm for Computational Problem Solving" in accordance with the Universities and University Colleges Act 1971 and the Constitution of the Universiti Putra Malaysia [P.U.(A) 106] 15 March 1998. The Committee recommends that the student be awarded the Master of Science.

Members of the Thesis Examination Committee were as follows:

Razali bin Yaakob, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Masrah Azrifah binti Azmi Murad, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Zulaiha Ali Othman, PhD
Associate Professor
Universiti Kebangsaan Malaysia
Malaysia
(External Examiner)

ZULKARNAIN ZAINAL, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 17 November 2015

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. The members of the Supervisory Committee were as follows:

Teh Noranis Binti Mohd Aris, PhD
Senior Lecturer
Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Md. Nasir Sulaiman, PhD
Associate Professor
Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

BUJANG BIN KIM HUAT, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

© COPYRIGHT UPM

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: _____ Date: _____

Name and Matric No.: Maryam Rajabi – GS34294

© COPYRIGHT UPM

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to

Signature: _____
Name of Chairman of Supervisory Committee: Dr Teh Noranis Binti Mohd Aris

Signature: _____
Name of Member of Supervisory Committee: Associate Professor Md. Nasir Sulaiman

© COPYRIGHT UPM

TABLE OF CONTENTS

		Page
ABSTRACT		i
ABSTRAK		iii
ACKNOWLEDGEMENTS		v
APPROVAL		vi
DECLARATION		ix
LIST OF TABLES		xiii
LIST OF FIGURES		xiv
CHAPTER		
1	**INTRODUCTION**	1
	1.1 Background	1
	1.2 Problem Statement	2
	1.3 Research Objectives	3
	1.4 Research Questions	3
	1.5 Research Scopes	4
	1.6 Research Contributions	4
	1.7 Organization of the Study	5
2	**LITERATURE REVIEW**	6
	2.1 Introduction	6
	2.2 Computer Programming	6
	2.3 Computational Problem Solving	6
	2.3.1 Programmers' Level of Expertise	7
	2.3.2 Computational Problem Solving Techniques	8
	2.4 Agents And Multi Agents	8
	2.4.1 Why Are Agents Useful?	10
	2.4.2 Intelligent Agents	11
	2.4.3 Java Agent Development (JADE)	12
	2.5 Information Extraction	13
	2.6 Problem Solving Environments	14
	2.6.1 L.E.C.G.O	14
	2.6.2 PASS	16
	2.6.3 Object-Karel	17
	2.6.4 WYSIWYC	18
	2.6.5 RAPTOR	19
	2.6.6 Jeliot 3	20
	2.6.7 FMAS	22
	2.7 Summary	28
3	**RESEARCH METHODOLOGY**	29
	3.1 Introduction	29
	3.2 Overall Methodology of the Study	29
	3.3 Identifying Problem	30

© COPYRIGHT UPM

	3.4	Propose an Agent-Based Model for CPS Using Prometheus Design Tool	30
	3.5	Developed a Prototype for the Proposed Agent-Based CPS Model and Cps Extraction Algorithm	32
		3.5.1 Develop the CPS Model in JADE Framework	32
		3.5.2 CPS Extraction Algorithm	33
	3.6	Experiments	33
		3.6.1 Group of Study	33
		3.6.2 Test of Normality and Reliability Analysis	33
	3.7	Evaluation	34
		3.7.1 Accuracy Test Using Several Sample Problems	34
		3.7.2 Statistical Analysis	34
	3.8	Summary	34
4	**DESIGN OF THE CPS MODEL**		36
	4.1	Introduction	36
	4.2	The Proposed CPS Tool	36
	4.3	Design Steps with PDT	37
		4.3.1 System Specification	37
		4.3.2 Architectural Design	43
		4.3.3 Detailed Design	45
	4.4	The Proposed Extraction Algorithm	48
	4.5	Summary	50
5	**EXPERIMENTS AND RESULTS**		51
	5.1	Introduction	51
	5.2	Jade Implementation	51
		5.2.1 Communication	52
		5.2.2 Experimental Results of Agent-Based Extraction Model	57
		5.2.3 Problems' Experimental Results	70
	5.3	Statistical Performance Evaluation	73
		5.3.1 Data Collection	73
		5.3.2 Instruments	73
		5.3.3 Data Analysis	74
		5.3.4 Test of Normality	74
		5.3.5 Reliability Analysis	75
		5.3.6 Demographic Profile of Respondents	75
		5.3.7 Research Questions Findings	77
	5.4	Summary	82
6	**CONCLUSION AND FUTURE WORKS**		83
	6.1	Conclusion	83
	6.2	Future works	83
REFERENCES			85
APPENDICES			95
BIODATA OF STUDENT			107
LIST OF PUBLICATIONS			108

© COPYRIGHT UPM

LIST OF TABLES

Table		Page
2.1	Summary of the investigated environments	25
5.1	PAC chart of problem1	64
5.2	IPO chart of problem1	65
5.3	PAC chart of problem11	68
5.4	IPO chart of problem11	69
5.5	Summary of the experimental results	71
5.6	Test of Normality for Understanding	74
5.7	Test of Normality for Efficiency	74
5.8	Test of Normality for Usability	74
5.9	Cronbach's Alpha interpretation rule	75
5.10	Cronbach's Alpha Reliability for CPS tool	75
5.11	Distribution of Participants' Gender	76
5.12	Distribution of Participants' Educational Level	76
5.13	Distribution of Participants based on Age	76
5.14	The range of level of CPS tool	77
5.15	Descriptive Statistics for understanding	78
5.16	Frequency table of understanding	78
5.17	Descriptive statistics for efficiency	79
5.18	Frequency table of efficiency	79
5.19	Descriptive Statistics for usability	80
5.20	Frequency table of usability	80
5.21	Frequency of the overall given scores	81

© COPYRIGHT UPM

LIST OF FIGURES

Figure		Page
2.1	Multi agent system	10
2.2	The structure of JADE	13
2.3	Diagrammatic representation of the general architecture of L.E.C.G.O	15
2.4	The general interface of activity space within L.E.C.G.O	16
2.5	Relationships among core entities in PASS	17
2.6	The main window of the Oject-Karel environment	18
2.7	Annotated snapshot of the ALVIS LIVE! environment	19
2.8	RAPTOR in action	20
2.9	User Interface of Jeliot 3	21
2.10	The proposed model for Jeliot 3	21
2.11	Architecture of FMAS	23
2.12	A sample output of the FMAS	23
3.1	Flowchart of the employed methodology	29
4.1	The proposed computational model of presented research work	37
4.2	The notation in PDT	37
4.3	Analysis overview diagram	39
4.4	Scenario overview diagram	39
4.5	Extract scenario	40
4.6	Transform scenario	41
4.7	Generate module number scenario	41
4.8	Goal overview diagram	42
4.9	System role overview diagram	43
4.10	Data coupling overview diagram	44
4.11	Agent role grouping overview diagram	44
4.12	System overview diagram	45
4.13	User_Agent overview diagram	46
4.14	PAC agent overview	47
4.15	IPO agent overview diagram	47
4.16	Algorithm agent overview diagram	48
4.17	Pseudo-code of extraction algorithm	49
5.1	A partial view of building message	52
5.2	A partial view of ‘User_Agent’ class	53
5.3	A segment of sending message	54
5.4	A partial view of sending message to PAC agent	54
5.5	A partial view of sending message to IPO agent	55
5.6	A partial view of sending message to Algorithm agent	55
5.7	The segment of receive message	56
5.8	The segment of receive message in CPS tool	56
5.9	The segment of update console and status box in the CPS tool	57
5.10	GUI remote agent management	58
5.11	Clicking ‘Read problem from txt file’ box	59
5.12	Browsing for the text file the stored problem	60
5.13	Illustration of the problem in ‘Enter the problem’ box	60

© COPYRIGHT UPM

5.14	Illustrating the event followed by problem transferring to the agent's action	61
5.15	ACL-message of User_Agent a) Problem1 b) Problem 11	61
5.16	PAC agent of problem 1	64
5.17	IPO agent of problem 1	65
5.18	Algorithm agent of problem 1	66
5.19	Algorithm structure of problem 1	66
5.20	PAC agent of problem 11	67
5.21	IPO agent of problem 11	68
5.22	Algorithm agent of problem 11	69
5.23	Algorithm structure of problem 11	70
5.24	Performance factors graph	81

© COPYRIGHT UPM

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, computer programming is known as a core basic skill, not only essential for those students studying in the computer science and software engineering disciplines, but also required in other technology disciplines including physical sciences as well as engineering (Xinogalos, 2015; Law et al., 2010; Lam et al., 2008). Programming problems are rooted from a huge number of problem domains. Therefore, comprehending the problem domain is of significance. Additionally, problem solving is generally addressed as the most crucial cognitive activity every human being deals with especially in the professional contexts (Gunbas, 2015; Pilli and Aksu, 2013; Sanchez and Olivares, 2011; Law et al., 2010; Chang et al., 2006; Jonassen, 2003).

This study is about agent-based model (ABM) which is also known as multi agent system (MAS) in computational problem solving. ABM is considered as a complex system in which a large number of intelligent and autonomous agents are interacting to fulfill a cooperative or an individual. An intelligent agent as the key component of many ABM, can be autonomous, proactive, reactive as well as sociable objective (Barriuso et al., 2015; Barbati et al., 2012; Chen, 2012; Lianzhong., 2008; Wooldridge and Jennings, 1995).

Moreover, based on one of the definitions of ABM considering it as a society of autonomous agents in which each one of them is able to act on its own, it can then be concluded that each agent has its own view of the world and therefore, no global control is applicable on ABM. Thus, in order to study ABM, one has to inevitably study the interactions amongst its components (autonomous agent) but not the behavior of any individual agent (Gao and Gu, 2009; Jennings, 2000). This approach, therefore, describes the system's behavior as a whole (emergence of the behavior). Therefore, a successful methodology of modeling ABM according to Schumacher (2001) is necessarily a behavioral-based or Bottom-up approach. Moreover, since ABM is viewed as an interaction amongst autonomous agents, the bottom-up approach implicitly implies managing the coordination of the agents.

In computer science, a pedagogical agent is of some specific learning objectives and 'acts' in educational settings in an autonomous way (Kadhim et al., 2014; Gulz, 2004). Accordingly, pedagogical agents can play many different roles in a learning environment and can facilitate learning in nearly any knowledge domain. For example, while some pedagogical agents can provide adaptive instruction to the learner through artificial intelligence (Moreno et al., 2001), not all pedagogical agents have these features (Schroeder and Adesope, 2014). Likewise, software agents are reactive systems and that are 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 (Gaur and Soni, 2015; Hosseinali et al.,

© COPYRIGHT UPM

2014; Gorodetsky, 2012). ABMs 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 (Barriuso et al., 2015; Sheng, 2014). Regarding the capability of agent-based systems, it should be noted that these systems usually have been utilized in industrial applications but merely have been used to help in designing of programming environments (Barbati et al., 2012; Liu et al., 2011; Hao et al., 2006).

In this study, the main focus is to investigate the possibility to use agent-based technology as a new agent model for computational problem solving to help novice programmers those who possess basic knowledge of programming.

1.2 Problem Statement

Learning to solve problems is too seldom required in formal educational settings, in part, because our understanding of its processes is limited. Instructional-design research and theory has devoted too little attention to the study of problem-solving processes.

The problem here is related to the difficulties of novices in understanding problem statements and transforming them to computational problem solving techniques (Hooshyar et al., 2015; Aris, 2012). It is very important to cater this problem from the beginning before writing the source codes. Due to these problems, researchers have tried to employ agent technology and Artificial Intelligence (AI) in order to promote intelligent tutoring systems. However, it is obviously difficult for a machine to understand the natural language as it usually does not follow fixed rule and structures. To have all these barriers in mind, it is vital to find a solution to create intelligent and autonomous systems to be able of helping teachers and students through the learning process. Agents have shown quiet good performance adding flexibility and autonomy to the intelligent systems. Agent-based systems can be used in order to solve the difficult or impossible problems for an individual agent or a monolithic system (Yang, 2014; Gorodetsky, 2012; Wooldridge, 2009; Padgham et al., 2008b).

Yet, to date, the proposed conventional computational problem solving tools have covered only some of the Computational Problem Solving (CPS) techniques such as flowchart, structure chart, algorithm and so forth. In all of these tools, visual interface has been employed to simplify the process of comprehension for novices (Hooshyar et al., 2014; Ben-Ari et al., 2013). The main issue which is associated with the aforementioned tools is that they assume that the students have already gained programming knowledge. Meanwhile, some other researches have been conducted utilizing animations or games to assist novice programmers in this matter (Bachu and Bernard, 2014; Sung and Hwang, 2013; Moreno et al., 2013; Ben-Ari et al., 2011). These tools also suffer from lack of utilization of the agent techniques while it could significantly improve the flexibility of the tools. As a matter of fact, software agents are capable of carrying out the things on behalf of the users based on the decisions they made. This is the main feature of an agent which derives decisions autonomously even in unexpected situations (Barriuso et al., 2015; Ralha and Silva, 2012).

© COPYRIGHT UPM

In this way, in terms of the agent-based computational problem solving tool, this thesis addresses the following problem statements (Hooshyar et al., 2015; Ben-Ari et al., 2013; Aris, 2012; Kris et al., 2010):

1. Novices generally suffer from lack of enough knowledge and comprehension in transforming the problem statements into problem solving techniques namely PAC, IPO and algorithm (Pseudo-code).
2. Because of the distinctively demanding requirement for learning computer programming, educators are highly needed to teach such courses aimed at identifying the set of motivating factors such as 'reward and recognition', 'individual attitude and expectation' and so on which give some incentives to their students' learning empirically and systematically.
3. Most of the proposed tools so far do not give the PAC, IPO and algorithm out of the given problem at once targeting at providing and guiding the novices in computational problem solving tools computations in detail.

Indeed, the above mentioned problems addressed some of the issues in CPS which have not been properly investigated by the aforementioned computational problem solving tools. Therefore, an extraction algorithm can be utilized in the proposed agent-based computational problem solving tool for carrying out the functions.

1.3 Research Objectives

The main objective of this research is to propose an agent-based model as a new agent tool for computational problem solving. In order to achieve this, the research objectives are as follow:

1. To propose an agent-based model using Prometheus methodology to design a new agent-based CPS tool.
2. To evaluate the effectiveness of the prototype in terms of understanding, efficiency and usability factors.

This mentioned modeling techniques propose a new computational tool of agent-based system which can be a complement result to the previously work in computer programming.

1.4 Research Questions

Based on the above mentioned objective of this study the following questions have been formulated:

1. How can an agent-based system be designed to cater for computational problem solving?
2. What is the information to be passed among agents?
3. How can agents be used in extraction?
4. Does extraction algorithm improve understanding of computational problem solving?
5. How to evaluate the effectiveness of the extraction algorithm and the tool?

© COPYRIGHT UPM

6. What are the benefits of the tool for problem solving skills based on users' perception?
    a. Visualization of operation of PAC
    b. Visualization of operation of IPO
    c. Visualization of operation of ALGORITHM (Pseudo-code)

1.5 Research Scopes

This research work focuses on designing and implementing a CPS tool with the help of agents (using agent). Computational problem solving is taken into account as the first step before source code development. The scenario is as follows: a simple problem or exercise given to students in text form is considered as a starting point. The proposed agent-based model comprises of four agents namely "User_Agent", "PAC" Agent, "IPO" Agent and "Algorithm" Agent. The problems consist of various examples in computer programming in text form which is divided into sequential and looping problems. Sequential problems are simple problems while looping problems are those which require repetitive or iterative process. The user Agent is the interface used by the user aimed at interacting with other agents. It starts from the document in text which is extracted by the PAC agent. The IPO agent extracts information from the PAC agent and then the algorithm agent extracts information from the IPO agent respectively. Indeed, such agents are at an intersection with each other via sending data including keywords, Input/Process/Output, I/P/O module number and process. Finally, the output of this model will be shown in the form of problem solving technique. It is expected that multi-agent based model is to reduce any programming errors or bugs while developing programs. In order to apply the practical reasoning agents in terms of the standard Belief, Desire and Intention (BDI) architecture, Java Agent Development (JADE) has been used for programming.

1.6 Research Contributions

In the research work, the existing conventional problem solving techniques are used with the goal of assisting novices in computer programming field. To this end, an agent based computational problem solving tool is proposed to detect and extract pre-requisite information out of computational problems in order to create PAC chart, IPO chart and ALGORITHM. In fact, the proposed tool is different from other visualization problem solving models in terms of employed architecture and algorithm. The defined extraction algorithm ought to be able of dealing with variety of computational problems including sequential problems and looping problems.

In this way, the contributions of this study are as follows:

1. A novel model by Prometheus Design Tool: To achieve this goal, a number of steps are involved for mapping the model to several diagrams namely Analysis Overview, Scenario Overview, Goal Overview diagram and System Overview, System Role Overview, Data Coupling Overview, Agent Role Grouping Overview, User_Agent, PAC, IPO and ALGORITHM.
2. A new tool to help students solve computational problems. The tool comprises of the extraction, transformation and generating module number tasks.

© COPYRIGHT UPM

a. The extraction task is consisted of following phases:
    - Text to PAC
    - PAC to IPO
    - IPO to Algorithm
b. The transformation task includes:
    - PAC to IPO
    - IPO to Algorithm
c. Generating module number is performed by IPO agent to dedicate unique numbers to each module.

3. A new agent-based model to improve understanding of computational problem solving, provide efficiency and usability of the tool.

1.7 Organization of the study

This thesis includes six chapters. A brief overview of each chapter and the organization of the thesis will be described in lines below.

The first chapter is a comprehensive introduction of the thesis. In this chapter, an overview of the thesis has been introduced through presenting a background for the study, problem statement, objective, scope and contribution of this research work. Finally, in this chapter the organization of this thesis and summary are illustrated.

Chapter 2, provides an overview to the previously reported works on the proposed multi-agent based computational problem solving model.

In Chapter 3, the proposed methodology of this thesis is introduced. The proposed methodology is defined through two computational modeling technique, based on Prometheus methodology (PDT) and agent technology. Moreover, an overview to the basic concept of the PDT and agent technology is represented in this chapter.

In Chapter 4, the design and analysis of the proposed agent-based model for computational problem solving environment is illustrated in details. Chapter 4 starts with introducing the proposed agent-based computational problem solving tool. Next, using Prometheus, the proposed agent-based tool has been modeled.

Chapter 5 contains the results of this thesis explained through the proposed software system for modeling the proposed agent-based computational problem solving tool. This chapter depicts on the developed extraction algorithm. It also demonstrates the inputs and outputs of the proposed agent based model. In addition, the accuracy of the computed results is discussed in this chapter. Finally, the obtained data from a questionnaire on the proposed tool functionality is analysed and discussed.

Chapter 6 presents a summary of the thesis the conclusion of this research. The objectives as well as the contributions of thesis is also revisited and recapped in this chapter. This final chapter is ended with casting lights on the ways aimed at improving this research and also the future works.

© COPYRIGHT UPM

REFERENCES

Ala-Mutka, Kirsti M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education, 15*(2), 83-102.

Anantharangachar, R., Ramani, S., & Rajagopalan, S. (2013). Ontology Guided Information Extraction from Unstructured Text. arXiv preprint arXiv:1302.1335. and evidence. *International Journal of Artificial Intelligence in Education, 14*, 313–334.

Aris, Teh Noranis Mohd. (2011). Object-Oriented Programming Semantics Representation Utilizing Agents. *Journal of Theoretical and Applied Information Technology, 31*(1).

Aris, Teh Noranis Mohd. (2012). *An Agent-Based Conceptual Model For Computational Problem Solving.* Paper presented at the The International Conference on Informatics and Applications (ICIA2012).

Bachu, E., & Bernard, M. (2014, January). *Visualizing Problem Solving in a Strategy Game for Teaching Programming.* In Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Barbati, M., Bruno, G., & Genovese, A. (2012). Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications, 39*(5), 6020-6028.

Barriuso, A., De la Prieta, F., & Li, T. (2015). An Agent-Based Social Simulation Platform with 3D Representation for Labor Integration of Disabled People. *In Trends in Practical Applications of Agents, Multi-Agent Systems and Sustainability* (pp. 55-64). Springer International Publishing.

Bednarik, R., Joy, MS, Moreno, A., Myller, N., Sun, S, & Sutinen, E. (2005). *Multi-Agent Educational System for Program Visualization*.

Bednarik, R., Moreno, A., & Myller, N. (2006). Program visualization for programming education–Case of Jeliot 3. *Association for Computing Machinery New Zealand Bulletin, 2*(2).

Bell, D.A., Beck, A., Miller, P., Wu, Q.X., and Herrera, A. (2007).Video Mining – Learning Patterns of Behaviour via an Intelligent Image Analysis System. *Proc. of the 7th International Conference on Intelligent Systems Design and Applications*, (pp. 460-464).

Bellifemine, F. L., Caire, G., & Greenwood, D. (2007). *Developing multi-agent systems with JADE* (Vol. 7): John Wiley & Sons.

© COPYRIGHT UPM

Bellifemine, F. L., Caire, G., Poggi, A., & Rimassa, G. (2008). JADE: A software framework for developing multi-agent applications. Lessons learned. *Information and Software Technology*, *50*(1), 10-21.

Ben-Ari, M., (2013). Visualization of Programming. *Improving Computer Science Education*, *52*.

Ben-Ari, M., Bednarik, R., Levy, R.B-B., Ebel, G., Moreno, A., Myller, N., and Sutinen, E. (2011). A Decade of Research and Development on Program Animation: The Jeliot Experience. *Journal of Visual Languages and Computing*, vol. 22, pp. 375-384.

Berners-Lee, Tim, Hendler, James, & Lassila, Ora. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.

Boyer, K.E., Lahti, W., Philips, R., Wallis, M.D., Vouk, M.A., & Lester, J.C. (2010), Principles of Asking Effective Questions during Student Problem Solving. *Proc. of the 41st ACM Technical Symposium on Computer Science Education*, pp. 460-464.

Burnett, C., Norman, T. J., & Sycara, K. (2013). Stereotypical trust and bias in dynamic multiagent systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *4*(2), 26.

Byrne,B. M.(2013). *Structural equation modeling with AMOS*:Basic concepts, and programming.

Carlisle, M. C., Wilson, T. A., Humphries, J. W., & Hadfield, S. M. (2005). *RAPTOR: a visual programming environment for teaching algorithmic problem solving.* Paper presented at the ACM SIGCSE Bulletin.

Carlisle, M. C. (2009). Raptor: a visual programming environment for teaching object-oriented programming. *Journal of Computing Sciences in Colleges*, *24*(4), 275-281.

Carolan, S., Chinesta, F., Evain, C., Magnin, M., & Moreau, G. (2014). Extracting domain ontologies from reference books. *In 2014 IEEE 14th International Conference on Advanced Learning Technologies (ICALT)* (pp. 544-545). IEEE.

Cetin, I. (2013). Visualization: a tool for enhancing students' concept images of basic object-oriented concepts. *Computer Science Education*, *23*(1), 1-23, DOI: 10.1080/08993408.2012.760903.

Chakraborty, S., & Gupta, S. (2014). Medical application using multi agent system-a literature survey. *International Journal of Engineering Research and Applications*, *4*(2), 528-546.

Chang, K. E., Sung, Y. T., & Lin, S. H. (2006). Computer assisted learning for mathematical problem solving. *Computers and Education*, *46*(2), 140–151.

© COPYRIGHT UPM

Chen, L. (2012). Agent-based modeling in urban and architectural research: A brief literature review. *Frontiers of Architectural Research, 1*(2), 166-177.

Choy, Marian, Lam, Sam, Poon, Chung Keung, Wang, Fu Lee, Yu, Yuen Tak, & Yuen, Leo. (2008). Design and implementation of an automated system for assessment of computer programming assignments. *Advances in Web Based Learning–ICWL 2007* (pp. 584-596): Springer.

Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, *59*(2), 661-686.

Dang, S., & Ahmad, P. H. (2014). Text Mining: Techniques and its Application. *International Journal of Engineering & Technology Innovations*, ISSN (Online), 2348-0866.

Dann, W. P., Cooper, S., & Pausch, R. (2011). *Learning to Program with Alice (w/CD ROM)*: Prentice Hall Press.

De Villiers, M. R., & Anniroot, J. (2012). *Study Of Alice: A Visual Environment For Teaching Object-Oriented Programming.*

DeLoach, S. A., & Garcia-Ojeda, J. C. (2010). O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems. *International Journal of Agent-Oriented Software Engineering, 4*(3), 244-280.

DeVellis, R.-F. (1991). *Scale development: Theory and applications*. Newbury Park, CA: Sage.

Dhavachelvan, P., Uma, G. (2005), Multi-agent-based integrated framework for intraclasstesting of object-oriented software, *Applied Soft Computing*, *5*(2), 205–222.

Eliassi-Rad, T., & Shavlik, J. (2003). A system for building intelligent agents that learn to retrieve and extract information. *User Modeling and User-Adapted Interaction, 13*(1-2), 35-88.

Gao, Y., & Gu, N. (2009). Complexity, human agents, and architectural design: a computational framework. *Design Principles and Practices: An International Journal, 3*(6), 115–126.

Gaur, V., & Soni, A. (2015). A knowledge-driven approach for specifying the requirements of multi-agent system. *International Journal of Business Information Systems*, *19*(3), 300-323.

Georgantaki, S., & Retalis, S. (2007). Using educational tools for teaching object oriented design and programming. *Journal of Information Technology, 2*, 111-130.

© COPYRIGHT UPM

George, D., & Mallery, P. (2001). *SPSS for windows: 10.0* Update. Massachusettes: Allyn & Bacon, 84-87.

Gorodetsky, V. I. (2012). Self-organization and multi-agent systems I. Models of self-organizing multi-agent. *Teoria i sistemy upravlenia*, *2*, 92-120.

Gulz, A. (2004). *Benefits of virtual characters in computer-based learning environments*: Claims.

Gunbas, N. (2015). Students' mathematics word problem-solving achievement in a computer-based story. *Journal of Computer Assisted Learning*, *31*(1), 78-95.

Guo, P. and Qi, H. (2010). *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.

Hair, J. F., Anderson, R. E., & Tatham, R. L. Black. WC, (2010). *Multivariate Data Analysis.A GLOBAL Perspective (7th Edition).*Upper Saddle River,New Jersey:Pearson Prentice Hall.

Hao, Q., Shen, W., Zhang, Z., Park, S.-W., & Lee, J.-K. (2006). Agent based collaborative product design engineering: An industrial case study. *Computers in Industry*, *57*, 26-38.

Henderson-Sellers, B., & Giorgini, P. (2005). *Agent-oriented methodologies*: IGI Global.

Hooshyar, D., Ahmad, R. B., Nasir, M. H. N. M., Shamshirband, S., & Horng, S. J. (2015). Flowchart-based programming environments for improving comprehension and problem-solving skill of novice programmers. a survey. *International Journal of Advanced Intelligence Paradigms*, *7*(1), 24-56.

Hooshyar, D., Ahmad, R. B., Nasir, M., & Mun, W. C. (2014, June). *Flowchart-based approach to aid novice programmers: A novel framework*. In Computer and Information Sciences (ICCOINS), 2014 International Conference on (pp. 1-5). IEEE.

Hooshyar, D., Ahmad, R. B., Shamshirband, S., Yousefi, M., & Horng, S. J. (2015). A flowchart-based programming environment for improving problem solving skills of Cs minors in computer programming.*The Asian International Journal of Life Sciences*, *24*(2), 629-646.

Hosseinali, F., Alesheikh, A. A., & Nourian, F. (2014). Rapid urban growth in the Qazvin region and its environmental hazards: implementing an agent-based model. *Polish Journal of Environmental Studies*, *23*(3).

Hui TK, Wan D and Ho A. (2007). Tourists' satisfaction, recommendation and revisiting Singapore. *Tourism Management, 28*, 965–975.

© COPYRIGHT UPM

Hundhausen, Christopher D, & Brown, Jonathan L. (2007). What You See Is What You Code: A "live" algorithm development and visualization environment for novice learners. *Journal of Visual Languages & Computing, 18*(1), 22-47.

Jenkins, T. (1998). *A participative approach to teaching programming* (Vol. 30, No. 3, pp. 125-129). ACM.

Jennings, Nicholas R. (2000). On agent-based software engineering. *Artificial intelligence, 117*(2), 277-296.

Johnson, R. A., & Wichern, D. W. (1998). *Applied multivariate statistical analysis.* USA: Prentice-Hall International Inc.

Jonassen, D. H. (2003). Designing research-based instruction for story problems. *Educational Psychology Review, 15*,267–296.

Kadhim, M. A., & Alam, M. A. (2012). To Developed Tool, an Intelligent Agent for Automatic Knowledge Acquisition In Rule-based Expert System. *International Journal of Computer Applications, 42*(9), 46-50.

Kadhim, M. A., Alam, M. A., & Kaur, H. (2014). A Multi-intelligent Agent Architecture for Knowledge Extraction: Novel Approaches for Automatic Production Rules Extraction. *International Journal of Multimedia and Ubiquitous Engineering, 9*(2), 95-114.

Kedar ,S. V.(2009). *Programming And Problem Solving.* Technical Publications.

Kerly, A., Ellis, R., & Bull, S. (2009). Conversational agents in E-Learning. *Proceedings of AI 2008* (pp.169–182).

Khue, N. T. M. (2012). *Developing an Intelligent Multi-Agent System based on JADE to solve problems automatically.* Paper presented at the International Conference on Systems and Informatics (ICSAI), 2012.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92*(1), 109-129.

Konecki, M., Kadoic, N., & Piltaver, R. (2015, May). *Intelligent assistant for helping students to learn programming.* In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on (pp. 924-928). IEEE.

Kordaki, Maria. (2006). *'Learning activity'as the basic structural element for the design of web-based content: A case study.* Paper presented at the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.

Kordaki, Maria. (2009). *Beginners' programming attempts to accomplish a multiple-solution based task within a multiple representational computer environment.*

© COPYRIGHT UPM

Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications.

Kordaki, Maria. (2010). A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation. *Computers & Education, 54*(1), 69-87. doi: http://dx.doi.org/10.1016/j.compedu.2009.07.012.

Krathwohl, D. R. (2002). A Revision of Bloom's Taxonomy: An Overview. *Theory into Practice, 41*(4), 212-218.

Kulkarni, A.J., & Tai, K. (2010). Probability collectives: A multi-agent approach for solving combinatorial optimization problems, *Applied Soft Computing, 10*(3), 759–771, http: //dx.doi.org/10.1016/j.asoc.2009.09.006.

Kumar, R., & Rosé, C. (2011). Architecture for building conversational agents that support collaborative learning. IEEE Transactions on Learning Technologies, 4(1), 21–34.

Law, K. M., Lee, V., & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. *Computers & Education, 55*(1), 218-228.

Leea, S., Yoonb, B., and Parkc, Y. (2009). An approach to discovering new technology opportunities: Keyword-based patent map approach, *Technovation, 29*, 81-497.

Leeton, U., & Kulworawanichpong, T. (2012, March). *Multi-Agent based optimal power flow solution.* In Power and Energy Engineering Conference (APPEEC), 2012 Asia-Pacific (pp. 1-4). IEEE.

Lewis, A. B., & Mayer, R. E. (1987). Students' miscomprehension of relational statements in arithmetic word problems. *Journal of Educational Psychology, 79*, 363–371.

Lianzhong, Liu, Xiangrong, Zu, & Ruzhi, Xu. (2008, 13-16 July 2008). *Multi-agent system coordination architecture and its use in electric power decision support system.* Paper presented at the 6th International Conference on Industrial Informatics (INDIN), 2008, IEEE.

Liu, C. C., Cheng, Y. B., & Huang, C. W. (2011). The effect of simulation games on the learning of computational problem solving. *Computers & Education, 57*(3), 1907-1918.

Liu, D. R., & Ke, C. K. (2007). Knowledge support for problem-solving in a production process: A hybrid of knowledge discovery and case-based reasoning. *Expert Systems with Applications, 33*(1), 147-161.

Lu, G., Lu, J., Yao, S., & Yip, Y. J. (2009). A review on computational trust models for multi-agent systems. *The open information science journal, 2*, 18-25.

© COPYRIGHT UPM

Luger, G. F. (2011). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Published by Dorling Kindersley (India) Pvt. Ltd., 5th Edition.

Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. *Instructional Science, 26*(1), 49–63.

Moghadam, M.K., Shojafar, M., Nami, M.R. and Rashidi, H. (2013). A new multi-agent system for e-health functionalities. *International Journal of Software Engineering and Its Applications, 7*(3), 23-34.

Moreno, A., Joy, M., & Sutinen, E. (2013). Roles of animation tools in understanding programming concepts. *Journal of Educational Multimedia and Hypermedia, 22*(2), 165-184.

Moreno, R., Mayer, R. E., Spires, H. A., & Lester, J. C. (2001). The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents? *Cognition and Instruction, 19*(2), 177–213.

Mousavi, A., Nordin, M. J., & Othman, Z. A. (2012). Ontology-driven coordination model for multiagent-based mobile workforce brokering systems. *Applied Intelligence, 36*(4), 768-787.

Odell, James J, Parunak, H Van Dyke, & Fleischer, Mitchell. (2003). Modeling agent organizations using roles. *Software and Systems Modeling, 2*(2), 76-81.

Omicini, Andrea, & Ossowski, Sascha. (2003). Objective versus Subjective Coordination in the Engineering of Agent Systems. In M. Klusch, S. Bergamaschi, P. Edwards & P. Petta (Eds.), *Intelligent Information Agents* (Vol. 2586, pp. 179-202): Springer Berlin Heidelberg.

Padgham, Lin, & Winikoff, Michael. (2003). Prometheus: A methodology for developing intelligent agents. *Agent-Oriented Software Engineering III* (pp. 174-185): Springer.

Padgham, Lin, & Winikoff, Michael. (2005). *Developing intelligent agent systems: A practical guide* (Vol. 13): John Wiley & Sons.

Padgham, Lin, Thangarajah, John, & Winikoff, Michael. (2008b). The prometheus design tool–a conference management system case study. *Agent-Oriented Software Engineering VIII* (pp. 197-211): Springer.

Pallant, J. (2010). *SPSS survival manual: A step by step guide to data analysis using the SPSS program.* (4th ed.). Maidenhead, Berkshire, England: McGraw-Hill International.

Pedrycz, W. and Rai, P. (2008). A Multifaceted Perspective at Data Analysis: a Study in Collaborative Intelligent Agents. *IEEE Trans. Systems, Man, and Cybernetics, 38*(4), 1062-1072.

© COPYRIGHT UPM

Pilli, O., & Aksu, M. (2013). The effects of computer assisted instruction on the achievement, attitudes and retention of fourth grade mathematics students in North Cyprus. *Computers and Education, 62*, 62–71.

Ralha C. G., & Silva C. V. S. (2012). A Multi-agent data mining system for cartel detection in Brazilian government procurement. *Expert Systems with Applications, 39*(14), 11642-11656.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.

Ropero, J., Gómez, A., Carrasco, A., & León, C. (2012). A Fuzzy Logic intelligent agent for Information Extraction: Introducing a new Fuzzy Logic-based term weighting scheme. *Expert Systems with Applications, 39*(4), 4567-4581.

Russell, S., & Norvig, P. (2003). Intelligence, Artificial. A modern approach. *Artificial Intelligence.* Prentice-Hall, Egnlewood Cliffs, 25.

Sanchez, J., & Olivares, R. (2011). Problem solving and collaboration using mobile serious games. *Computers & Education, 57*, 1943–1952.

Savitha, R., & Porkodi, R. (2015). An Overview of Text Mining Techniques and Methodologies Used in Bioinformatics Domain.

Schmitz, M., Bart, R., Soderland, S., & Etzioni, O. (2012, July). *Open language learning for information extraction.* In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 523-534). Association for Computational Linguistics.

Schroeder, N. L. & Adesope, O. O. (2014). A Systematic Review of Pedagogical Agents' Persona, Motivation and Cognitive Load Implications for Learners. *Journal of Research on Technology in Education, 46*(3), 229–251.

Schumacher, Michael. (2001). Objective Coordination in Multi-Agent System Engineering: Design and Implementation. *Multi-agent systems*, 9-32.

Sekaran, U. (2003). *Research Methods for Business: A Skill Building Approach*. John Willey and Sons, New York.

Sharma, D., Trivedi, A., Srinivasan, D., & Thillainathan, L. (2013). Multi-agent modeling for solving profit based unit commitment problem. *Applied Soft Computing, 13*(8), 3751-3761.

Sivasakthi, M., & Rajendran, R. (2011). Learning difficulties of object-oriented paradigm using Java: Students' perspective. *Indian Journal of Science and Technology, 4*, 983–985.

© COPYRIGHT UPM

Su, C. J., & Wu, C. Y. (2011). JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring. *Applied Soft Computing, 11*(1), 315-325.

Sung, H. Y., & Hwang, G. J. (2013). A collaborative game-based learning approach to improving students' learning performance in science courses. *Computers & Education, 63*, 43-51.

Sycara, K. P. (1998). Multiagent systems. *AI magazine*, *19*(2), 79.

Sycara, K., Paolucci, M., Ankolekar, A., & Srinivasan, N. (2011). Automated discovery, interaction and composition of semantic web services. *Web Semantics: Science, Services and Agents on the World Wide Web*, *1*(1).

Sycara, K., Paolucci, M., Soudry, J., & Srinivasan, N. (2004). Dynamic discovery and coordination of agent-based semantic web services. *Internet Computing, IEEE, 8*(3), 66-73.

Tabachnick, B. G., & Fidell, L. S. (2007). *Using multivariate statistics* (5th ed.) (Vol. 3). Boston Pearson Education.

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*: MIT press.

Wooldridge, M. J. (2009). *An introduction to multiagent systems*: John Wiley & Sons.

Wooldridge, M. J. (1992). *The logical modelling of computational multi-agent systems* (Doctoral dissertation, University of Manchester, Institute of Science and Technology).

Wooldridge, M. J., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review, 10*(02), 115-152.

Xinogalos, S. (2013, March). *Using flowchart-based programming environments for simplifying programming and software engineering processes*. In Global Engineering Education Conference (EDUCON), 2013 IEEE (pp. 1313-1322). IEEE.

Xinogalos, S. (2014). Designing and deploying programming courses: Strategies, tools, difficulties and pedagogy. *Education and Information Technologies*, 1-30.

Xinogalos, S. (2015). Object-Oriented Design and Programming: An Investigation of Novices' Conceptions on Objects and Classes. *ACM Transactions on Computing Education (TOCE), 15*(3), 13.

Yang, S. Y. (2014). *A Web Services-Based and Ontology-Supported Case-Based Reasoning Information Agent.* Paper presented at the Proceedings of the 2014 International Symposium on Computer, Consumer and Control.

© COPYRIGHT UPM

Zheng, R., Xu, Y., Chakraborty, N., & Sycara, K. (2014, May). *Multiagent coordination for demand management with energy generation and storage*. In Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems (pp. 1587-1588). International Foundation for Autonomous Agents and Multiagent Systems.

Zhuo, L. (2011). *A multi-agent based approach for solving the redundancy allocation problem.* Temple University.

Zuo, X., Zhang, Z., & Xie, J. (2015). Detection on Large Amount of Web Pages Based on a Highly Active Way of Site-Level Template. *International Journal of Hybrid Information Technology*, *8*(4), pp. 259-266. http://dx.doi.org/10.14257/ijhit.2015.8.4.28.

© COPYRIGHT UPM