INTELLIGENT AGENT BASED PAIR PROGRAMMING AND INCREASED SELF-EFFICACY THROUGH PRIOR-LEARNING FOR ENHANCED LEARNING PERFORMANCE

L.Jegatha Deborah[1], R.Baskaran[2] and A.Kannan[3], P.Vijayakumar[4]

[1, 4]Department of Computer Science and Engineering, UCET, Melpakkam, Tindivanam, India-604 001
[2]Department of Computer Science and Engineering, Anna University, Chennai, India – 600 025
[3]Department of Information Science and Technology, Anna University, Chennai, India – 600 025
E-mail: [1]blessedjeny@gmail.com, [2]baaski@annauniv.edu, [3]akannan@annauniv.edu, [4]vijibond2000@gmail.com

ABSTRACT

Performances of the students in learning a programming course is not same, since learning to program is greatly influenced by two dominating factors namely self-efficacy and mental efforts. Prior research efforts have shown that high self-efficacy can have an increased effect of being a trained programmer, especially in an intelligent agent based pair programming system. The main objective of this work is to increase the self-efficacy of the students by providing prior-learning experiences. This experience is facilitated by recommendation agents that provide suitable E-Learning programming course contents based on identifying their individual learning styles which can be used as a factor of prior self-learning computing experience. This helps in increasing the programming abilities when learning in an agent-based pair programming environment subsequently. Moreover, the proposed system analyzes the educational effects of the students learning using pair programming agents based on increased self-efficacy.

Keywords: E-Learning, Pair programming, Self-Efficacy, Learning Styles, E-Learning content, Recommendation Agents

1.0 INTRODUCTION

Several factors affect the performance in learning a programming course [1] and research is under progress to identify why most of the students drop out from learning the programming courses. Learning a programming course is comparatively difficult than learning any other course in E-learning since, programming knowledge requires cognitive abilities, logical thinking and mastering the abstract concepts [2], [51], [58]. Therefore, this situation is a significant problem in the field of Computer Science and Engineering. There are several studies which address these problems by providing many methods of educational systems as indicated in Table 1. Prior research has shown that effective factors like improving self-efficacy and good metal effort can increase the performance of the students, especially when learning a programming course [3], [57].

Table 1 Different Types of Educational Systems

Types of Educational Systems	Related Work – Authors and Year of Publication
Offline systems	M. Bari and B. Lavoie (2007), N.P.Pearson and A.C.Graesser (2006), K. Han et al (2010)
Web-based systems	D. Palmieri (2002), H. Ueno (2000), P.Woods and J.Warren (1995), H. J. S. de Azevedo and E. E. Scalabrin (2005)
Intelligent systems	V. Devedzic (2004), E. Sklar and D. Richards (2006), N. D .Fleming (2001)
Adaptive system	G. W. Dekker et al (2009), R. E .Boyatzis, & D. A .Kolb (1997), J. Cohen (1998)
Performance Evaluations	D. Martinez (2001), N. Myller et al (2002), G. Chen et al (2000)

Several factors influence the success of novice programmers namely Prior Computing Experiences [4-5], E-Learning [6], Computer Entertainment during Training [7], Self-Efficacy [8], knowledge management [50], Collaborative learning Environments [9], [55 – 56], Learning Styles [10-12], [54] and Students Mental Effort [13-14], [57]. Among the above mentioned factors, providing a collaborated learning environment with increased self-efficacy of the students is the main focus of this work which leads to the success of the students in learning programming courses [48].

Pair programming is a term used in learning environments where two programmers work together at one workstation on the same design, algorithm or code [15], [49], [56]. Since, at least two people are involved, this is a form of collaborating learning environment. This kind of collaboration is found beneficial to the students especially when learning a programming course. Mastering a programming course requires greater effort provided by the students in addition to several cognitive factors like mental effort, self-efficacy and pre-training [16].

Self-efficacy is a key component in learning activities for learning involves more than just acquiring skills. Bandura [17] defines self-efficacy as people’s own judgment of their capabilities to organize and execute the courses of actions required to achieve a specific goal. Educational researchers recognize that, because skills and self-beliefs are so intertwined, one way of improving student performance is to improve student self-efficacy [18]. Attempts have also been made, with some success, to increase self-efficacy in learning by peer modeling of tasks, verbal persuasion, or other types of social influences, such as cooperative learning environments [17 – 20], [54]. Persons who possess higher self-efficacy belief show more effort and resistance for completing tasks and therefore have better and more effective task-fulfillment compared with individuals who have weak self-efficacy [20]. However, increasing the self-efficacy by the above methodologies is found to be inefficient for the students learning through the web. According to Bandura's theory, self-efficacy belief is under the influence of the following elements:

- ✓ Personal experience which leads to success or failure
- ✓ Observing the behavior of the model, the substitute, or the Example
- ✓ Vocal encouragement
- ✓ Considering the physiological conditions

In such a scenario, the self-efficacy of the students can be increased if they are allowed to judge their own abilities, before they could pair to learn with appropriate agents to master a programming course. Moreover, as analyzed before the students could outperform well in learning a programming course, if they had prior programming experience by learning the basic concepts [21]. This is achieved, by allowing the students to self-learn the E-contents from E-Learning servers based on their choice of learning materials available in different formats namely documents, audio and video lectures. This increases the level of self-efficacy because the target students acquire some basic knowledge about the programming course. A considerable increase in the self-efficacy can make them to outperform well in the subsequent learning with peer-learning agents in a Pair Programming strategy [16]. The main objective of the proposed system is to increase the self-efficacy of the students by allowing them to self-learn using recommended E-Learning contents based on their learning styles and then master the programming course using peer-learning agents in a Pair Programming strategy.

The rest of the paper is organized as follows. Section 2 provides a summary of the related work in Intelligent Tutoring Systems. Section 3 depicts the proposed system architecture. Section 4 presents the working of the architectural components. Section 5 provides the methodology of implementation. Section 6 details the experimental results and the subsequent discussions. Section 7 provides the concluding remarks with some of the references provided at the end.

2.0 RELATED WORK

The traditional offline educational system is neither intelligent nor adaptive in nature [22]. In such systems, the instructional materials are delivered to the students through dialogues and interactions. The next era of educational system is intelligent tutoring systems. In these systems, the face-to-face interactions are replaced by e-mails, chats and online discussion forums. Several Artificial Intelligence techniques are used in such systems to provide personalized teaching and learning process [23], [53], [57].

An agent-based learning system can be used in tutoring systems to provide assistance and to monitor the performance of those students [24], [52]. Animated pedagogical agents interact with students in a manner that closely resembles face-to-face collaborative learning [25]. Successful Collaborative learning approaches have five critical attributes namely a common task, small group learning, cooperative behavior, interdependence and individual accountability [26]. Prior researches have shown that pair programming when used as a collaborative learning positively affects student knowledge retention and knowledge transfer [27]. In such an environment, knowledge transfer is achieved by switching roles between a student and an agent. Interactive teaching and learning by a student and an agent achieves better performance in learning a programming course. Among the several merits of pair programming, there are some demerits listed in the study. One of the main drawbacks is that when students try to master a particular programming course from the scratch, the effect of self-efficacy of the students plays a major role. In such a scenario, it is very mandated to increase the self-efficacy of the students and this is achieved by making them to self-learn using suitable E-Learning contents in E-Learning servers based on their learning styles.

The INTELLITUTOR system attempts to teach novice Pascal and C programmers as a human programming tutor [28]. The system is capable of detecting logical errors and provides the types and location of bugs in the form of feedbacks. However, the system is not capable of understanding the students' self-efficacy which helps in determining their success and failure rate.

RAPTIS is an intelligent programming tutor system that teaches Pascal looping concepts [29]. This system is adaptive in nature where a variety of teaching strategies are employed and the strategies could be changed according to the students' history. However, this system does not aim to increase the self-efficacy of the students. Increasing the self-efficacy might be a permanent solution for an effective teaching-learning methodology.

The agents developed by Silva de Azevedo and Scalabrin require other agents or humans to support collaborative learning [30]. The roles defined by their agents are to help a human perform innovative tasks, to support dynamic interactions with students and to act as a teacher with clear and objective information about the students. In the midst of these, the agents fail to understand the students' capabilities to perform a task effectively and to increase their capabilities during teaching and learning.

Pedagogical agents can be designed for personalization and adaptive. Pearson et al developed pedagogical agents that facilitate students learning by supporting collaborative, interactive and investigative learning in web-based learning technologies [25]. The agents in these systems merely guide a learner through a lesson using computer-assisted interactions. However, these agents have responsibilities in neither determining their self-efficacy nor increasing it.

Agents developed by Devedzic facilitate students' motivation and provides adaptive feedbacks in E-Learning environments [31]. They are designed in such a way to assist the learning process in different domains. But, these agents could not increase the students' self-efficacy in terms of identifying their learning styles.

Three types of agents namely pedagogical, peer-learning and demonstrating agents help to assist the human learners which were developed by Sklar et al [32]. Pedagogical agents interact with a learner and monitors the students performance indirectly which helps to understand the student and provides feedbacks. Peer-Learning agents acts like peers and are interactive partners but are less engineered than pedagogical agents. Demonstrating agents are agent-based simulations or educational robotics and act like interactive mediums for learning. However, this system could be enhanced by introducing new kinds of agents which could identify the students' self-efficacy through the prediction of learning styles. Moreover, it could be increased by providing suitable E-Learning contents as a pre-requisite for learning with peer-learning agents. The peer-learning agents developed by Han et al [16] provide more positive impact on students achievements and their self-efficacy. The determined self-efficacy is not found to be increased by offering suitable E-Learning contents for pre-tutoring to the students based on their learning styles.

Han et al [26] developed an agent system and analyzed the educational effects of a peer-learning agent based on pair programming strategy. In this system, the agents and the students switched their roles of 'learning by doing' and 'learning by teaching'. This system attempted to enhance their learning through the use of the peer-learning agents. In this system, the students are judged for their capabilities by making the student to study the basic concepts of programming with the peer agents and appearing for a pre-test. In such a scenario, the psychological level of the learners is not well balanced since the learning styles of the learners vary from one individual to another. The success of the E-Learning system may degrade if a similar kind of E-contents is provided to all the students. The students can learn the basics of programming courses from the E-contents based on their choices namely documents, audio and video. This kind of E-contents provision can increase their self-efficacy and the students can now learn with the help of peer-learning agents to master the programming course. Table 2 provides the summary of the related works discussed based on the impact of self-efficacy.

Table 2. Summary of Related Works of Agents based on Self-Efficacy

Agent System	Proposed Year	Agents Participation in addressing Self-Efficacy
INTELLITUTOR	2000	• Not addressed
RAPTIS	1995	• Students past history is identified for content delivery
Silva de Azevedo and Scalabrin agents	2005	• self-efficacy is addressed in the form collaborative learning • But failed to increase it
Pedagogical agents	2006	• Not addressed
Devedzic agents		• provides motivation in the form of feedbacks • But failed to increase it
Pedagogical, Peer-learning and demonstrating agents	2004	• Not addressed
Han et al agents	2007	• students achievements are monitored
Han et al agents	2010	• Self-efficacy in addressed in the form of providing collaborative teaching and learning using peer-learning agents • Does not handle about increasing self-efficacy to enhance the performances
Proposed Agent system	2012	• Self-efficacy is identified through predicting the students learning styles • Also aims at increasing the self-efficacy by recommending suitable E-contents to self-learn to increase the performance of them when learning with peer-learning agents subsequently

From the discussions made on the intelligent tutoring systems, it is identified that increasing the self-efficacy can enhance the performance of the students in E-Learning of programming courses. This paper analyzes in increasing the self-efficacy of the students by identifying their learning styles. Identifying the learning styles of the students can help in facilitating them with suitable recommendations on E-contents available in E-Learning servers. Moreover, Flemming VAK learning style model is used in identifying the learning styles of the students [33]. After identifying their learning styles, they are recommended with suitable E-contents based on their choice of learning. This helps them to self-learn the basic concepts of C programming course. Self-learning the basic concepts of C programming language helps in better collaborative learning with peer-learning agents which helps in mastering the programming course.

3.0 PROPOSED SYSTEM ARCHITECTURE

The use of the Internet in education has provided a new revolution known as E-Learning or web-based learning [34]. In spite of all the important contributions provided by various researchers in the area E-Learning described in Table 1, it will be useful if machine learning techniques are used to further improve the self-efficacy of the students involved in learning. Moreover, it is proved that the exchange of roles and the meaningful feedback between the peer-learning agent and the student have a positive effect on learning [16, 26]. Moreover, the peer-learning agents enhance the effect of learning by determining the students' level of understanding. However, increasing the self-efficacy of the students can still enhance the performance of the students [3]. Prior literatures have suggested several factors of increasing the self-efficacy of the students in terms of providing previous computing experiences through short lectures or self-learning, computer entertainment during training, motivations, learning styles and assessing the students mental model during learning. This paper focuses on the recommendations of suitable E-Learning contents available in E-Learning servers to self-learn. This is done in order to facilitate the students with previous computing experiences to understand the basic concepts in a particular programming course. Moreover, suitable E-Learning contents are provided based on identifying their learning styles using Flemming VAK learning style model [33]. Subsequently, the students are paired with peer-learning agents for mastering the programming courses. The students are assessed for their performances during self-learning and learning with peer-learning agents. The proposed system architecture is shown in Fig.1.

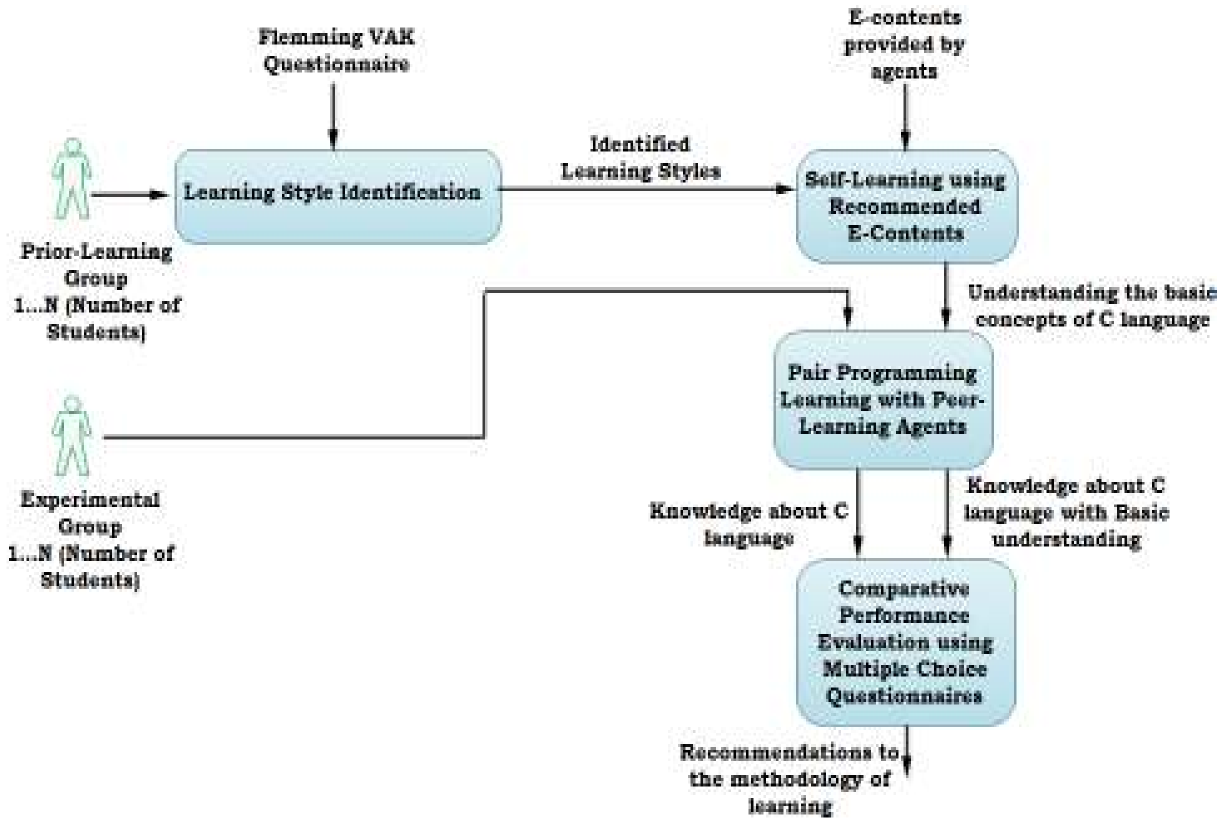

Fig.1. Proposed System Architecture

4.0 ARCHITECTURE WORKING COMPONENTS

4.1 Learning Style Identification

The students involved in E-Learning develop high interest in learning web contents posted in any E-Learning servers. A wide variety of E-Learning servers are available like Moodle, MediaWiki and Joomla. We used MediaWiki server for testing the proposed system. The students are identified for their individual learning styles based on Flemming VAK learning style model inventory [33]. According to the model, the students were categorized for three types of learning which are defined below.

- ✓ Visual – People with a visual learning style have a preference for seeing or observed things, including pictures, diagrams, demonstrations, displays, handouts, films and flip charts.
- ✓ Auditory – People with an auditory learning style have a preference for the transfer of information through listening to the spoken word, self or others, sounds and noises.
- ✓ Kinaesthetic – People with a kinaesthetic learning style have a preference for physical experience of touching, feeling, holding, doing and practical hands-on experiences.

The target course of learning in this work is C programming language. The C programming language contents are posted in various formats like documents, audio files and video lectures. These contents are made available in MediaWiki E-Learning server. The students are now analyzed for their learning styles using Flemming VAK learning style questionnaires. These questionnaires are evaluated to classify the students into three kinds of students having specific learning styles namely visual, auditory and kinaesthetic. In such a scenario, the recommendation agents available in the system can provide some recommendations to the students on the choice of E-learning contents for self study based on their learning styles. For instance, students with visual learning styles can be recommended to learn the basic concepts of C programming language available in video lectures, students with an auditory learning style can be recommended to learn C programming available in audio lectures and students with kinaesthetic learning styles can be recommended to learn the same available in documents and exe files for hands-on experiences.

4.2 Self-Learning

The students after proper authentication are allowed to access the E-Learning server for any type of contents to learn C programming language. The students can access any type of contents like documents, audio and video lectures.

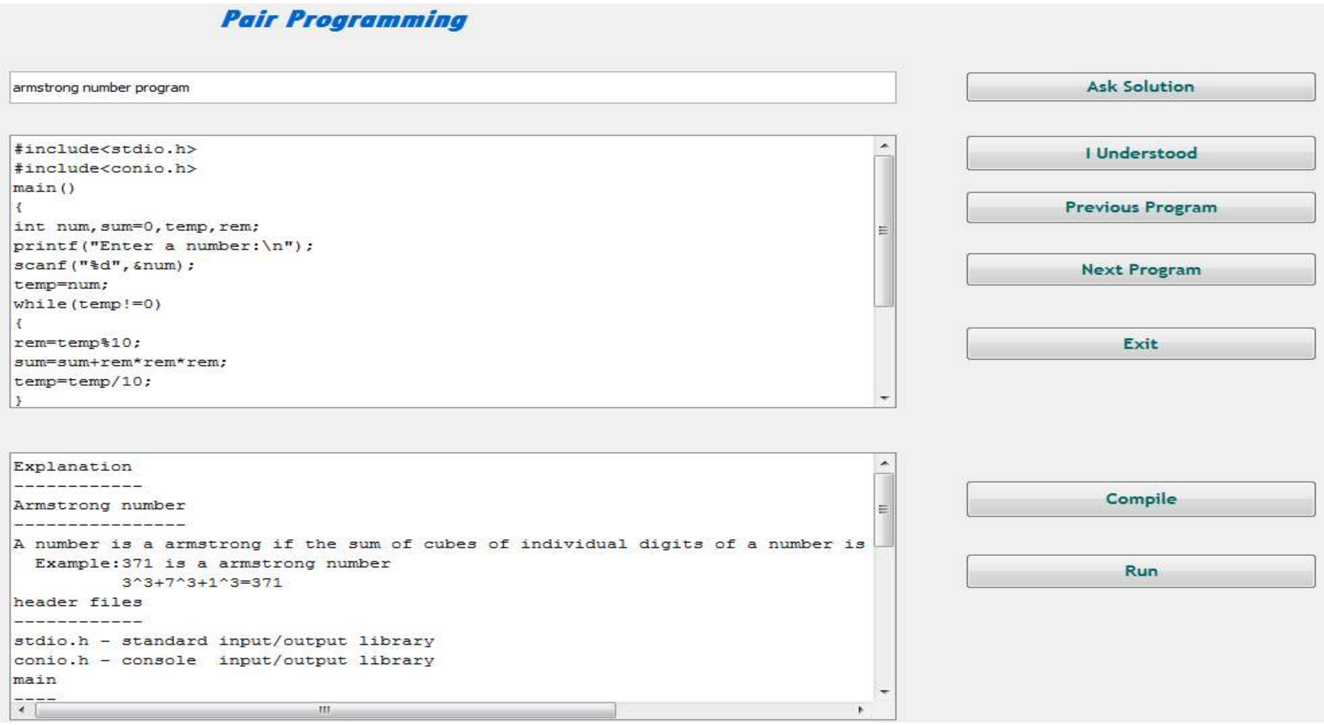

Fig.2. Agent Platform Window: Student : Tutee, Agent : Tutor

Since, the students are provided with a variety of contents, they can be given some kind of recommendations to learn a particular content so that their self-efficacy can be increased. This is evident from the results presented in section 6 where the students' performance of mean value is higher than the students with no prior knowledge of their self-efficacy. In connection to this, they are suggested for a specific content of C programming course based on their identified learning styles. The students are given 4 weeks of time in order to learn the basic concepts of C programming language like data types, identifiers, operators and looping statements. After 4 weeks of self–learning, the students gain some knowledge of the basic concepts of C language. This helps in improved self-efficacy in order to master C programming language course. The students can be tested for their learning experiences, but however, this test is not treated to be mandatory.

4.3 Learning with Peer-Learning Agents

The students have some basic knowledge about the C programming language course. This aids in increasing their self-efficacy, since they have been given pre computing experiences of learning contents based on the identified learning styles. The peer-learning agents and the students switch between the roles of tutor and tutee. Since, the students have obtained the basic knowledge about C programming language, they learn with the peer-learning agents on advanced programming. We tested our proposed system for teaching and learning C programs like Armstrong number, Factorial number and the Fibonacci series. The students initiate the system by providing the logic of the program and the peer-learning agent helps the students in writing the syntactically correct program. In this case, the student assumes the role of tutee and the agent assumes the role of a tutor. For the next time, the peer-learning agents explain the logic of the code to the students and they teach the agents in writing the syntactically correct program. In this case, the students and agents assume the roles of tutor and tutee respectively. In this kind of mechanism, the students and the agents alternate the roles of tutor and tutee and master the C programming language course. This teaching-learning process is done for 4 weeks of time as given for self-learning. After 4 weeks of time, the students are again assessed for their performances using a test consisting of multiple-choice questions. Their performances are evaluated for comparison. Fig.2 depicts the snapshot of the agent window when the students' acts as a tutee and the agents' acts like a tutor. Fig.3 shows the screenshot of the agent platform window when the agent acts like a tutee and the student acting like a tutor.

5.0 METHODOLOGY

This study sought to determine the impact of giving prior-learning experiences by providing suitable E-Learning contents based on their learning styles was beneficial to the students for subsequent learning with peer-learning agents. The self-efficacy of the students is judged by their learning styles [35]. This efficacy is increased by providing suitable E-contents to the students for self-learning based on their learning styles. The system was evaluated using 86 students from four Engineering Departments of Anna University and this is given by the variable 'N' in Tables 3 and 4.

All these students were interested in learning C programming language course. Forty-three students from two departments were assigned to a group called prior-learned group, where these students were provided with self-learning E-contents recommended based on their learning styles. The remaining 43 students from the other two departments were assigned to a group called experimental group. The prior-learned the group had a self-learning experience through recommended learning using E-contents posted by E-learning servers based on their learning styles. The experimental group did not have self-learning experience and had no choice of assessing their self-efficacy and learnt C programming language directly using peer-learning agents. The performance assessments of both the groups were tested for knowledge retention and knowledge transfer, eight questions for each of the categories. The pre test and post test consisted of 16 questions (8 on retention and 8 on transfer) which were reviewed for effectiveness by a professor of Anna University considered as the domain expert in the C programming language course.

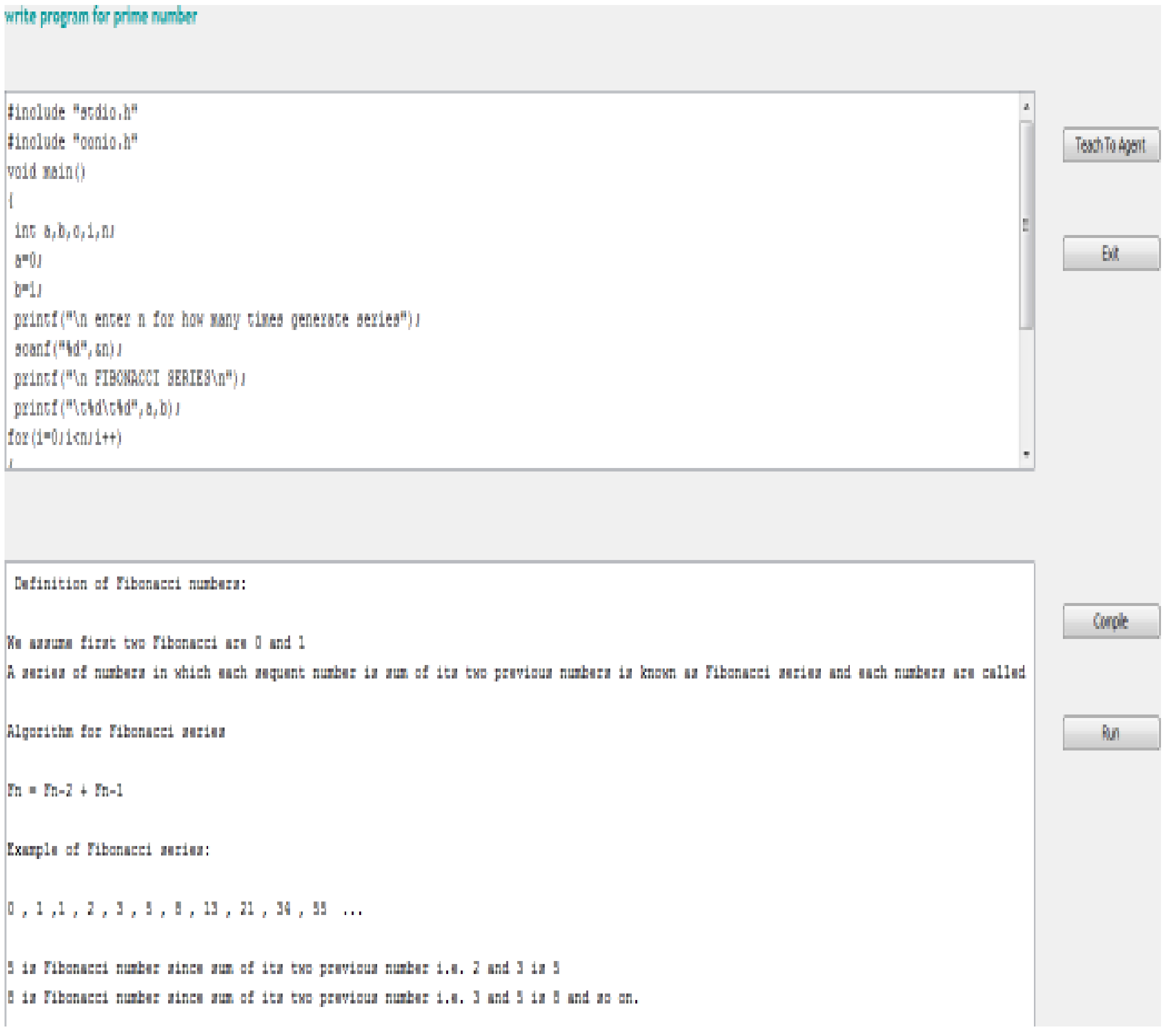

Fig.3. Agent Platform Window: Agent : Tutee, Student : Tutor

6. 0 RESULTS AND DISCUSSIONS

6.1 Learning Styles Identification

The students are varied in nature and the same kind of E-content delivery will not have any impact on subsequent learning. Their self-efficacy is judged using their learning styles identification. It is a proven fact that increasing the self-efficacy helps in creating a positive atmosphere in learning which increases the performance of the students exponentially. The proposed system aims at increasing the self-efficacy of the students by providing suitable C programming E-contents available in documents, audio and video lectures in MediaWiki E-learning server. The learning styles are identified using Flemming VAK learning style inventory questionnaires. Subsequent to the identification of learning styles, they are recommended to learn the E-contents available in E-learning servers based on their learning styles. This increases the self-efficacy of the students substantially which helps in mastering the C programming course when learning with peer-learning agents subsequently.

Fig.4 depicts the screen shot of the recommendation agent available in the system showing recommendations on the E-contents available in E-Learning servers based on their learning styles which increases their self efficacy.

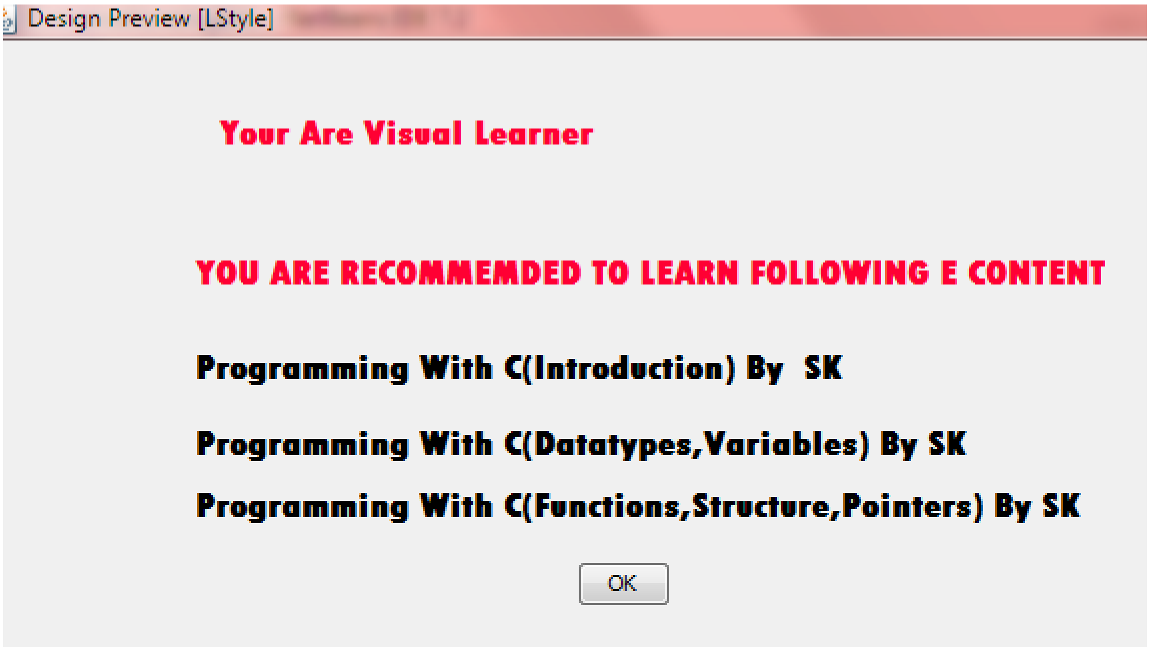

Fig.4. Recommendation Agent System Platform

6.2 Pre-Test

The system conducted a pre-test for both self-learned group and experimental group. This test was conducted to see if the two groups were homogeneous in terms of the levels of achievement. The test consisted of 16 questions (8 on knowledge retention and 8 on knowledge transfer). The t-test results of the pre-test are shown in Table 3. The results found in Table 3 indicate that the groups had some significant differences in their performances. But according to the results of the pre-test on retention and transfer, there was no statistically much difference between the self-learned group and the experimental group and the difference is found to be greater than 0.05.

6.3 Post-Test

After 8 weeks of time for teaching and learning C programming language course using self-learning and peer-learning agent strategy, the students are evaluated for their performances. The test consisted of questionnaires as done earlier. Now, the groups are separated widely. The self-learned the group had pre-learning experience for 4 weeks of time using the recommended E-contents available in E-Learning servers based on their learning styles. They underwent paired programming learning strategy with peer-learning agents for another 4 weeks of time. However, the experimental group learnt with the peer-learning agents for 8 weeks of time without having a pre-learning experience. Both the groups are evaluated using a test of 16 questions (8 on knowledge on retention and 8 on transfer). The t-test results of the post-test are shown in Table 4. According to the results shown in Table 4, there was a large difference which is statistically significant between the prior-learned and experimental groups and the values are found to be less than 0.05. The statistical difference for Knowledge Retention between prior-learned group and experimental group is 0.0035 and the statistical difference for Knowledge Transfer between the same groups is 0.0006. Moreover, it is also evident from Table 4 that, the mean scores of the prior-learned group is greater than the experimental group. Students who had a self-learning experience using the recommended E-contents from E-Learning servers based on their learning styles had very high performance results compared to the students who directly learnt with the peer-learning agents using a pair programming strategy. The results of Table 4 indicate that increased self-efficacy of the students along with pair programming strategy using peer-learning agents had a positive effect on knowledge retention and transfer in C programming language.

Table 3. Pre-test Evaluations on the metrics of Knowledge retention and transfer using t-test

Metrics	Group	N	Mean	Standard Deviation	t-test values	Significant difference
Knowledge Retention	Prior-Learned	43	56.50	25.60	1.9215	0.416
	Experimental	43	55.50	25.32		
Knowledge Transfer	Prior-Learned	43	50.80	16.07	0.7385	0.542
	Experimental	43	50.40	16.67		

Table 4. Post-test Evaluations on the metrics of Knowledge retention and transfer using t-test

Metrics	Name of the Groups	N	Mean	Standard Deviation	t-test values	Significant difference
Knowledge Retention	Prior-Learned	43	76.10	17.94	3.9489	0.0034
	Experimental	43	69.70	14.93		
Knowledge Transfer	Prior-Learned	43	88.30	5.77	5.1604	0.0006
	Experimental	43	66.40	12.95		

6.4 Discussions

It has been proved by Han et al [26] that the students learning with peer learning agents performed well than the student learning with traditional agents. However, we have enhanced the works done by Han et al and proved in this paper that the performance of the students is evaluated by considering the self-efficacy of the students in terms of providing self-learning experiences. In this paper, the self-efficacy of the students is judged by identifying their learning styles. The self-efficacy of the individual students is increased by recommending suitable E-contents available in E-learning servers namely documents, audio and video lectures based on their learning styles. This kind of recommendation facilitates the students in pre-learning experiences. Subsequently, the students learn with peer-learning agents using a pair programming strategy. Learning with peer-learning agents are a form of collaborative learning where the student and the agent can alternate the roles of a tutor and a tutee. In this case, pair programming is used as a teaching-learning strategy. Hence, such exchange of ideas between the student and the agent can have a positive effect on teaching and learning. Moreover, the students have increased self-efficacy of having pre-learning experience based on their choice of E-contents. According to Cohen [36], the effect size of retention was 0.38, which is a medium effect size, whereas the effect size of transfer was 0.79, which is significant. The table indicates that the effect size of transfer exceeded that of retention. The proposed E-learning system was found to have a positive effect on teaching and learning strategies. Moreover, it is evident from Table 4 that prior learned group did extremely well compared to the experimental group in post test, since the prior learned group consisted of students who had self-learning experience on the basic concepts of C programming language using the recommended E-contents available in E-Learning servers based on their learning styles.

7. 0 CONCLUSION

This paper focuses on the impact of self-efficacy of the students which plays a key role in performance enhancement in E-Learning environments. Self-efficacy has been addressed by identifying the learning styles of the students using Flemming VAK learning style model. In addition, the self-efficacy is also increased by making the students to self-learn the recommended E-contents available in MediaWiki E-learning servers. It is observed that, students learning in a Pair Programming strategy using peer-learning agents after self-learning through E-contents which are recommended to them based on their learning styles perform better than students paired directly with peer-learning agents. Further works in this direction involve the incorporation of varied teaching and learning formats namely documents, audio and video lectures.

REFERENCES

[1] L. White and M. P. Sivitanides, "A theory of the relationships between cognitive requirements of computer programming languages and programmer's cognitive characteristics," *J. Inf. Syst. Educ.*, Vol. 13, No. 1, 2002, pp. 59–66.

[2] M. J. Van Gorp and S. Grissom, "An empirical evaluation of using constructive classroom activities to teach introductory programming," *Comput. Sci. Educ.*, Vol. 11, No. 3, 2001, pp. 247–260.

[3] Vennila Ramalingam, Deborah LaBelle, Susan Wiedenbeck, "Self-Efficacy and Mental Models in Learning to Program", In Proc. *Innovation and Technology in Computer Science Education*, NY.USA, pp. 171-175, 2004.

[4] E.D .Bunderson & M.E .Christensen, "An analysis of retention problems for female students in university computer science programs", *Journal of Research on Computing in Education*, Vol.28, No.1, 1995, pp. 1-15.

[5] H.Taylor & L.Mounfield, "Exploration of the relationship between prior computing experience and gender on success in college computer science", *Journal of Educational Computing Research*, Vol.11, No.4, 1994, pp. 291-306.

[6] L.Jegatha Deborah, R.Baskaran and A.Kannan, "Construction of Ontology using Computational Linguistics for E-Learning", In: Proceedings of the *Second International Conference on Visual Informatics*, Kualalumpur, Malaysia: Springer – Lecture Notes in Computer Science, pp. 50 -63, 2011.

[7] D.Potosky, "A field study of computer efficacy beliefs as an outcome of training: the role of computer playfulness, computer knowledge, and performance during training", *Computers in Human Behavior*, 18, 2002, pp. 241-255.

[8] P .Byrne, & G .Lyons, "The effect of student attributes on success in programming", ITiCSE: Proceedings of the *6th annual conference on Innovation and technology in computer science education*, ACM Press, NY, 2002, pp. 49-52.

[9] H. J. S. de Azevedo and E. E. Scalabrin, "A human collaborative online learning environment using intelligent agents in Designing Distributed Learning Environments with Intelligent Software Agents," Hershey, PA: Information Science Publishing, 2005, pp. 1–32.

[10] L .Thomas, J .Woodbury, & E .Jarman, "Learning styles and performance in the introductory programming sequence", Proceedings of *Technical Symposium on Computer Science Education* 2002 (Covington KY), ACM Press, NY, pp. 33-37, 2002.

[11] K. Butler, "A Learning and teaching style in theory and practice", Columbia, CT: Learner's Dimension.1986.

[12] L.Jegatha Deborah, R.Baskaran and A.Kannan, "Learning styles assessment and theoretical origin in an E-learning scenario: a survey", *Springer-Artificial Intelligence Review*, DOI 10.1007/s10462-012-9344-0 [Online First Article]

[13] E Soloway, & K.Ehrlich, "Empirical studies of programmer knowledge", *IEEE Transactions of Software Engineering*, Vol.10, No.5, 1984, pp. 595-609.

[14] S. Wiedenbeck, , V.Ramalingam, , S.Sarasamma, & C.L.Corritore, "A comparison of the comprehension of object-oriented and procedural programs by novice programmers", *Interacting with Computers*, 11, pp. 255- 282, 1999.

[15] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald, "Pair programming improves student retention, confidence, and program quality," *Commun. ACM*, Vol. 49, No. 8, 2006, pp. 90–95.

[16] K. Han, E. Lee, and Y. Lee, "The effects of a peer agent on achievement and self-efficacy in programming education," *J. Korea Assoc. Comput. Educ*, Vol. 10, No. 5, 2007, pp. 43–51.

[17] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change", *Psychological Review*, Vol.84, No.2, 1977, pp. 191-215.

[18] D.R.Compeau, & C.A.Higgins, "Computer self-efficacy: development of a measure and initial test", *MIS Quarterly*, Vol.19, No.2, 1995, pp. 189-211.

[19] J.J.Canas , M.T.Bajo, & P.Gonzalvo, "Mental models and computer programming", *International Journal of Human-Computer Studies*, Vol.40, No.5, 1994, pp. 795-811.

[20] M. E.Gist, C.Schwoerer, and B.Rosen, "Effects of alternative training methods on selfefficacy and performance in computer software training", *Journal of Applied Psychology*, Vol.74, 1989, pp. 884- 891.

[21] P. Kinnunen and L. Malmi, "Why students drop out CS1 course?," in Proceedings of *International Conference on Computational and Education Research*, 2006, pp. 97–108.

[22] J. E. Beck, R. Baker, A. T. Corbett, J. Kay, D. J. Litman, T.Mitrovic, and S. Ritter, "Analyzing Student tutor Interaction Logs Improve Educ. Outcomes", In Proceedings of the 7th International Conference, ITS 2004, Alagoas, Brazil, Vol. 3220, 2004, pp. 531 - 538.

[23] R.G. Raj and S. Abdul-Kareem. "A Pattern Based Approach for the Derivation of Base Forms of Verbs from Participles and Tenses for Flexible NLP". *Malaysian Journal of Computer Science,* 2011, 24(2): 63 – 72.

[24] M. Bari and B. Lavoie, "Predicting interactive properties by mining educational multimedia presentations," In Proc. *Int. Conf. Inf. Commun. Technol*, 2007, pp. 231–234.

[25] N.P.Pearson and A.C.Graesser, "Pedagogical agents and tutors", *In Encyclopaedia of Education*, J.W.Guthrie, Ed. New York: Macmillan, 2006

[26] K. Han, E. Lee, and Y. Lee, "The Impact of a Peer-Learning Agent Based on Pair Programming in a Programming Course," *IEEE Transactions on Education*, Vol. 53, No.2, 2010, pp. 318–327.

[27] D. Palmieri, "Knowledge management through pair programming," Master's thesis, Dept. Comput. Sci., North Carolina State Univ., Raleigh, NC, 2002

[28] H. Ueno, "A generalized knowledge-based approach to comprehend pascal and C programs," *IEICE Trans. Inf. Syst*., Vol. E81-D, No. 12, 2000, pp. 1323–1329.

[29] P.Woods and J.Warren, "Rapid prototyping of an intelligent tutorial system", In Proceedings of *12th Australian Society for Computers in Learning in Tertiary Education*, 1995, pp. 557-563.

[30] H. J. S. de Azevedo and E. E. Scalabrin, "A human collaborative online learning environment using intelligent agents in Designing Distributed Learning Environments With Intelligent Software Agents," Hershey, PA: Information Science Publishing, pp. 1–32, 2005.

[31] V. Devedzic, "Web intelligence and artificial intelligence in education", *Educ. Technol*. Soc., Vol. 7, No. 4, 2004, pp. 29–39.

[32] E. Sklar and D. Richards, "The use of agents in human learning systems," in Proceedings of *5th International Joint Conference on Autonomous Agents and MultiAgent Systems*, 2006, pp. 767–774.

[33] N. D .Fleming, "Teaching and learning styles: VARK strategies", Christchurch, New Zealand: N.D. Fleming. 2001.

[34] G. W. Dekker, M. Pechenizkiy, and J. M. Vleeshouwers, "Predicting students drop out: A case study," In Proceedings of *International Conference on Educational Data Mining*, Cordoba, Spain, 2009, pp. 41–50.

[35] R. E .Boyatzis, and D. A .Kolb, "Assessing individuality in learning: The Learning Skills Profile", *Educational Psychology*, Vol.11, No.3-4, 1997, pp. 279-295.

[36] J. Cohen, Statistical Power Analysis for the Behavioral Sciences (2nd ed.). Hillsdale, NJ: Earlbaum, 1988.

[37] D. Martinez, "Predicting student outcomes using discriminant function analysis," In Proceedings of *Annual Meeting of Research and Planning Group*, Lake Arrowhead, CA, 2001, pp. 1–22.

[38] N. Myller, J. Suhonen, and E. Sutinen, "Using data mining for improving web-based course design," In Proceedings of *International Conference on Computer Supported Education*, Washington, DC, 2002, pp. 959–964.

[39] G. Chen, C. Liu, K. Ou, and B. Liu, "Discovering decision knowledge from web log portfolio for managing classroom processes by applying decision tree and data cube technology," *J. Educ. Comput. Res*, Vol. 23, No. 3, 2000, pp. 305–332.

[40] V. Robinet, G. Bisson, M. Gordon, and B. Lemaire, "Searching for student intermediate mental steps," In Proc. *Int. Conf. User Model. Workshop Data Mining User Model.*, Corfu, Greece, 2007, pp. 101–105.

[41] J. Stamper and T. Barnes, "Unsupervised MDP value selection for automating ITS capabilities," In Proc. *Int. Conf. Educ. Data Mining*, Cordoba, Spain, 2009, pp. 180–188

[42] R. M. Crespo, A. Pardo, J. P. Perez, and C. D. Kloos, "An algorithm for peer review matching using student profiles based on fuzzy classification and genetic algorithms," In Proceedings of *International Conference on Innovative Application of Artificial Intelligence*, Bari, Italy, 2005, pp. 685–694.

[43] H. Ba-Omar, I. Petrounias, and F. Anwar, "A framework for using web usage mining for personalise e-learning," In Proceedings of *International Conference on Advance Learning Technology*, Niigata, Japan, 2007, pp. 937–938

[44] David Adrian Sanders and Jorge Bergasa-Suso, "Inferring Learning Style from the Way Students Interact With a Computer User Interface and the WWW", *IEEE Transactions on Education*, Vol. 53, No.4, Nov.2010, pp. 613 – 620.

[45] J. Bergasa-Suso, D. A. Sanders, and G. E. Tewkesbury, "Intelligent browser-based systems to assist Internet users," *IEEE Trans. Educ.*, Vol. 48, No. 4, 2005, pp. 580–585.

[46] L. Burr and D. H. Spennemann, "Pattern of user behavior in university online forums", *Int. J. Instruct. Technol. Distance Learn*. Vol. 1, No. 10, 2004, pp. 11–28.

[47] S. Zheng, S. Xiong, Y. Huang, and S.Wu, "Using methods of association rules mining optimization in web-based mobile learning system," In Proc. *International Symposium Electronic Commerce Security*, Guangzhou, China, 2008, pp. 967–970.

[48] Andersson Annika, Hedstrom Karin and Gronlund Ake, "Learning from eLearning: Emerging Constructive Learning Practices", Proceedings of the *International Conference on Information Systems*, Paper 51, pp. 75-95, 2009.

[49] Cesar Vialardi, Javier Bravo and Alvaro Ortigosa, " Improving AEH Courses through Log Analysis", *Journal of Universal Computer Science*,Vol. 14, No. 17, 2008, pp. 2777-2798.

[50] Chein, M. and Mugnier, M. "Graph-based Knowledge Representation", Computational Foundations of Conceptual Graphs, Springer, 2009.

[51] Dekker, G.W., Pechenizkiy, M. and Vleeshouwers, J.M. "Predicting students drop out: A case study", In Proceedings of *International Conference on Educational Data Mining*, Cordoba, Spain, pp. 41-50, 2009.

[52] Mahkameh Yaghmaie and Ardeshir Bahreininejad, "A context-aware adaptive learning system using agents", *Elsevier Journal of Expert Systems with Applications*, Vol. 38, No. 2, 2011, pp. 3280-3286.

[53] Honglei Zhuang, Jie Tang, Wenbin Tang, Tiancheng Lou, Alvin Chin and Xia Wang, "Actively learning to infer social ties". *Data Mining and Knowledge Discovery*, Vol. 25, No. 2, 2012, pp. 270-297.

[54] Jo Hannay,E., Erik Arisholm, Harald Engvik and Dag Sjoberg, I.K. "Effects of Personality on Pair Programming", *IEEE Transactions on Software Engineering*, Vol. 36, No. 1, 2010, pp. 61-80.

[55] Kim, H. K. and Bateman, B. L. "Student participation patterns in online discussion: Incorporating constructivist discussion into online courses". *International journal on e-learning*, Vol. 9, No. 1, 2010, pp. 79-98.

[56] Lui, K.M., Keith Chan, C.C. and John Teofil Nosek, "The Effect of Pairs in Program Design Tasks", *IEEE Transactions on Software Engineering*, Vol. 34, No. 2, 2008, pp. 197-211.

[57] L. P. Ranatunga , "On computing mental states", *Malaysian Journal of Computer Science*, Vol.9, No.2, 1996, pp. 43 – 53.

[58] Hameed. N,Cheah, Yu-N,Rafie, M, "XEBPER: an E-book using Java 3D API", *Malaysian Journal of Computer Science*, Vol.24, No.1, 2011, pp. 54 – 62.

BIOGRAPHY

1. ***Dr.L.Jegatha Deborah*** completed her Bachelor's degree in the Department of Computer Science and Engineering under Madurai Kamarajar University, Madurai during the year 2002. Following that, she pursued her Masters degree in the same discipline under Anna University Chennai during 2003-2005. She also completed her doctoral program in the branch of Information and Communication Engineering under Anna University Chennai in the year 2013. Her areas of expertise are Artificial Intelligence, Knowledge Engineering, Data Mining and Natural Language Processing.
2. ***Dr. R. Baskaran*** received M.E. degree from University of Madras in the year 2011 and Ph.D. degree from Anna University, Chennai-600 025, Tamil Nadu-India in the year 2007. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai-600 025, Tamil Nadu, India. His areas of interest include Database Systems, Data Mining, Artificial Intelligence and Multimedia
3. ***Dr. A. Kannan*** received M.Sc. degree from Annamalai University-Tamil Nadu in 1986, M.E. and Ph.D. degrees from Anna University, Chennai-600 025, Tamil Nadu-India in 1991 and 2000 respectively. He is currently working as a Professor in the Department of Information Science and Technology, Anna University, Chennai-600 025, Tamil Nadu. His areas of interest include Database Systems, Data Mining and Artificial Intelligence.