An Active Learning strategy Think-Pair$_{\text{Free Open Source Software}}$-Share to Teach Engineering Cources

Dr. S. A. Halkude,
Civil Engineering Department,
Walchand Institute of Technology,
Solapur, India.
halkude60@gmail.com

Mrs. Sunita M. Dol,
Computer Science and Engineering Department,
Walchand Institute of Technology,
Solapur, India.
sunita_aher@yahoo.com

Abstract— **TPS (Think-Pair-Share) is a cooperative learning strategy where students think about their responses for a problem given by instructor then discuss their individual solutions in pairs and share those solutions with the class. Think-Pair$_{\text{Free-Open-Source-Software}}$-Share (TP$_{\text{FOSS}}$S) is a modified activity of TPS in which free open source software can be used in TPS activity. How these two activities can be used in teaching the engineering courses is explained in this paper. Here, the case study of Theory of Computation which is the core course of Computer Science and Engineering and the base for many other courses like System Programming, Compiler Construction, etc. is considered. These TPS and TP$_{\text{FOSS}}$S activities are employed for the Theory of Computation course to improve students' conceptual understanding. In this paper, one group Pre-Test Post-Test model is considered. Experimental results, students' perception about these activities and effectiveness of these activities are also presented in the study.**

Keywords— ***Think-Pair-Share(TPS), Think-Pair$_{FOSS}$-Share (TP$_{FOSS}$S), Theory of Computation (TOC), t-Test.***

I. Introduction (*Heading 1*)

Effective and innovative teaching methods are always desirable to teach various engineering courses. Think-Pair-Share (TPS) is cooperative learning strategy which has been recommended for its benefits of allowing students to express their reasoning, reflect on their thinking, and obtain immediate feedback on their understanding [6], [7]. TP$_{\text{FOSS}}$S which is modified version of TPS is explained for Compiler Construction Course of Third Year Computer Science & Engineering [8] for UG programme. TPS and TP$_{\text{FOSS}}$S activities are employed for teaching the Theory of Computation course in the present study.

'Theory of Computation (TOC)' is one of the core course for Computer Science & Engineering Under Graduate (UG) programme and is the prerequisite for many other courses like System Programming, Compiler Construction, etc. Since it is core course it is needless to emphasise the need of clear understanding of the concepts of this course. This course is important for modelling different kinds of hardware and software. Also, this course is of paramount importance for competitive entrance examination for admissions to their postgraduate (PG) studies.

The research question in the present study is "whether the use of TPS and TP$_{\text{FOSS}}$S activities in class helps the students to improve the conceptual understanding of the course". To find the answer to this research question, one group pre-test and post-test experimental study along with feedback of students.

II. Literature Survey

There are numerous software tools available to teach this course [4], [5]. Additionally Mukta Goyal and Shelly Sachdeva in "Enhancing Theory of Computation Teaching through Integration with other Courses" [1] aimed towards introducing different approaches for making the course interactive and realistic by integrating it with other courses learnt in previous semesters and current semester of engineering [1].

Carlos I. Ches˜nevar et al. in "Didactic Strategies for Promoting Significant Learning in Formal Languages and Automata Theory" [2] introduced a number of didactic strategies based on a constructivist approach.

Carlos Iv´an Ches˜nevar et al. in "Teaching Fundamentals of Computing Theory: A Constructivist Approach" [3] were proposed a strategies based on a stronger use of technology and a constructivist approach.

III. Methodology

The research questions examined in this study are –

- How to make student to understand concept of finite automata, regular expression etc.?
- How to improve student's performance in examination?
- How to make student to understand the concepts of converting one form of machine to another, converting regular expression to NFA, converting grammar to normal forms, etc.?

A. *Sample Used*

A group of 40 students was selected for this experiment.

B. What is FOSS?

Free and open-source software (FOSS) is computer software. It can be classified as both free software and open source software. Anyone is freely licensed to use, copy, study, and change the software in any way. The source code is openly shared to user so that all concerned are encouraged to voluntarily improve the design of the software [9]. Here, for Theory of Computation, JFLAP free open source software is considered.

JFLAP: It is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, multi-tape Turing machines, several types of grammars, parsing, and L-systems, etc[10].

C. Organization of Case Study

To test the effectiveness of TPS and TPFOSSS, one group Pre-Test and Post-Test model is considered for Theory of Computation course. Simplified forms and normal forms are the topics considered for this activity.

First, the instructor taught this topic by traditional teaching method i.e. blackboard teaching method in the class. In this subtopics covered were - eliminating null production, eliminating unit production, eliminating the useless variable from the given context free grammar and finally converting the given context free grammar to Chomsky Normal Form. Pre-test covering these subtopics was conducted for 30 marks. After conducting the pre-test, instructor considered TPS activity in the classroom. For both TPS and TP$_{FOSS}$S activities, the problem statement was to convert the given context free grammar to Chomsky Normal form using following steps:

Step 1: Eliminate null productions from given context free grammar if any.

Step 2: Eliminate unit productions from given context free grammar if any.

Step 3: Eliminate useless variable from the productions of given context free grammar if any.

Step 4: Convert the context free grammar to Chomsky Normal Form.

TPS activity for converting the given context free grammar to Chomsky Normal form consist of

Think: In think phase of TPS activity, instructor asked the question to students to eliminate null productions, unit productions and useless variables, if any, from the given context free grammar.

Pair: In pair phase, each student was asked to pair with the partner, shared their thinking with each other and proceeds with the task. Instructor asked the question related previous one that is suitable to deepen the students' understanding of the topic. The students were asked to convert the grammar obtained in 'Think' phase to Chomsky Normal form.

Share: In share phase, students shared the solution with the entire class. Instructor discussed the problem of converting context free grammar to Chomsky Normal form and highlighted important points.

The TPS activity is shown in Figure 1. Two TPS activity was conducted for two different grammars. After TPS activity, post-test of 30 marks was conducted.

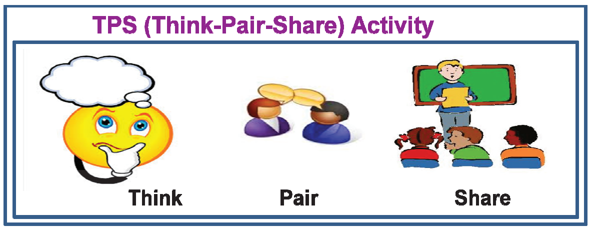


Fig. 1: TPS

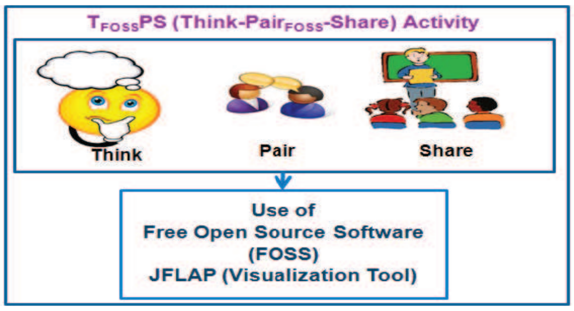

T$_{FOSS}$PS (Think-Pair$_{FOSS}$-Share) Activity
Think
Pair
Share
Use of
Free Open Source Software
(FOSS)
JFLAP (Visualization Tool)

Fig. 2: TP$_{FOSS}$S

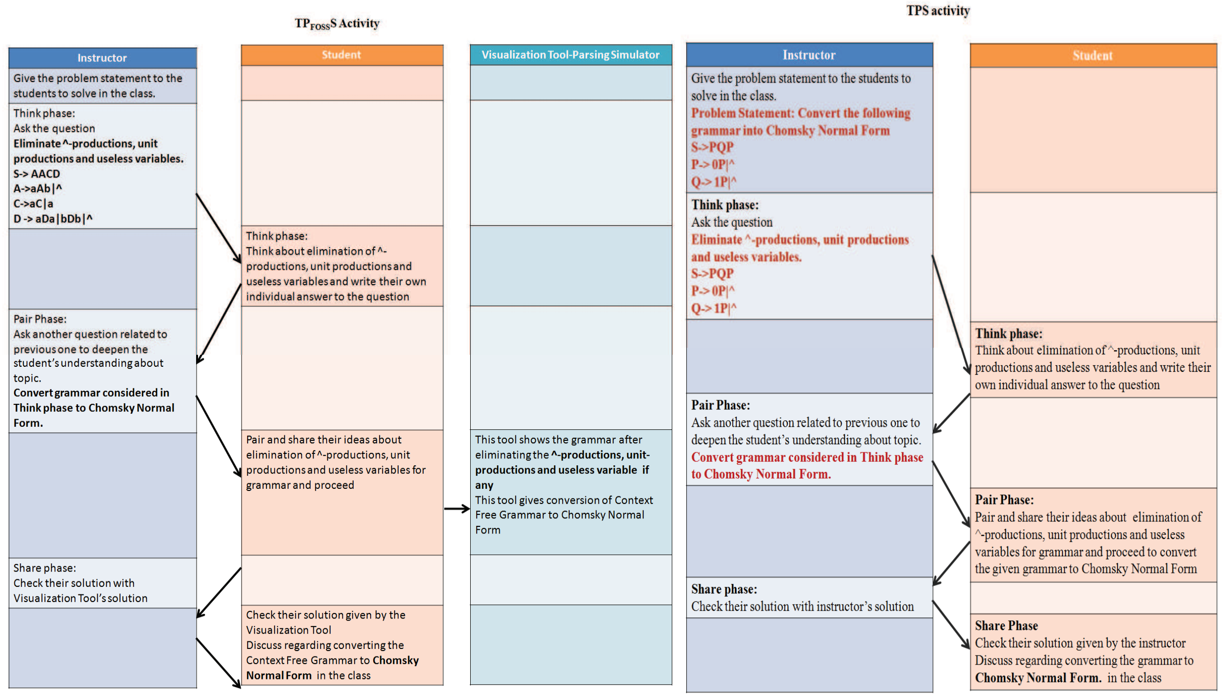
Two TP$_{FOSS}$S activity is conducted for two different examples. This activity differs from TPS in pair phase only. The TP$_{FOSS}$S activity is shown in figure 2. The pair phase of TP$_{FOSS}$S is explained as follows:

Pair$_{FOSS}$ (P$_{FOSS}$): In pair phase, each student was asked to pair with the partner student, shared their thinking with each other regarding eliminating null variable, unit production and useless variables for given context free grammar. Instructor displayed the solution after eliminating null variable, unit production and useless variables from given context free grammar using the tool-JFLAP.

Instructor asked the question related to previous one that is suitable to deepen the students' understanding of the topic. The students were asked to convert the grammar obtained in 'Think' phase to Chomsky Normal Form. Students were given time to convert the grammar to Chomsky Normal Form. After converting the grammar to Chomsky Normal Form, instructor displayed the solution of converting the grammar to Chomsky Normal Form to students using the tool JFLAP as shown in figure 3.

After TPFOSSS activity, one more post-test was conducted. To assess the effect of this activity on long term performance, the test after one month was conducted.

TPS and TPFOSSS activity example is given in the following Figure 3.


TP$_{FOSS}$S Activity

Instructor

Give the problem statement to the students to solve in the class.

Think phase:
Ask the question
Eliminate ^-productions, unit productions and useless variables.
S-> AACD
A->aAb|^
C->aC|a
D -> aDa|bDb|^

Pair Phase:
Ask another question related to previous one to deepen the student's understanding about topic.
Convert grammar considered in Think phase to Chomsky Normal Form.

Share phase:
Check their solution with Visualization Tool's solution

Student

Think phase:
Think about elimination of ^-productions, unit productions and useless variables and write their own individual answer to the question

Pair and share their ideas about elimination of ^-productions, unit productions and useless variables for grammar and proceed

Check their solution given by the Visualization Tool
Discuss regarding converting the Context Free Grammar to **Chomsky Normal Form** in the class

Visualization Tool-Parsing Simulator

This tool shows the grammar after eliminating the **^-productions, unit-productions and useless variable if any**
This tool gives conversion of Context Free Grammar to Chomsky Normal Form

TPS activity

Instructor

Give the problem statement to the students to solve in the class.
Problem Statement: Convert the following grammar into Chomsky Normal Form
S->PQP
P-> 0P|^
Q-> 1P|^

Think phase:
Ask the question
Eliminate ^-productions, unit productions and useless variables.
S->PQP
P-> 0P|^
Q-> 1P|^

Pair Phase:
Ask another question related to previous one to deepen the student's understanding about topic.
Convert grammar considered in Think phase to Chomsky Normal Form.

Share phase:
Check their solution with instructor's solution

Student

Think phase:
Think about elimination of ^-productions, unit productions and useless variables and write their own individual answer to the question

Pair Phase:
Pair and share their ideas about elimination of ^-productions, unit productions and useless variables for grammar and proceed to convert the given grammar to Chomsky Normal Form

Share Phase
Check their solution given by the instructor
Discuss regarding converting the grammar to **Chomsky Normal Form.** in the class

Fig.3: TPS and TP$_{FOSS}$S activity example.

The organization of case study is shown in Figure 4. The objectives of this study are:

- To understand the concepts of finite automata, regular expression etc.
- To improve the performance of students in this course.
- To make the concepts easy like converting one form of machine to another, converting regular expression to NFA, converting grammar to normal forms etc.

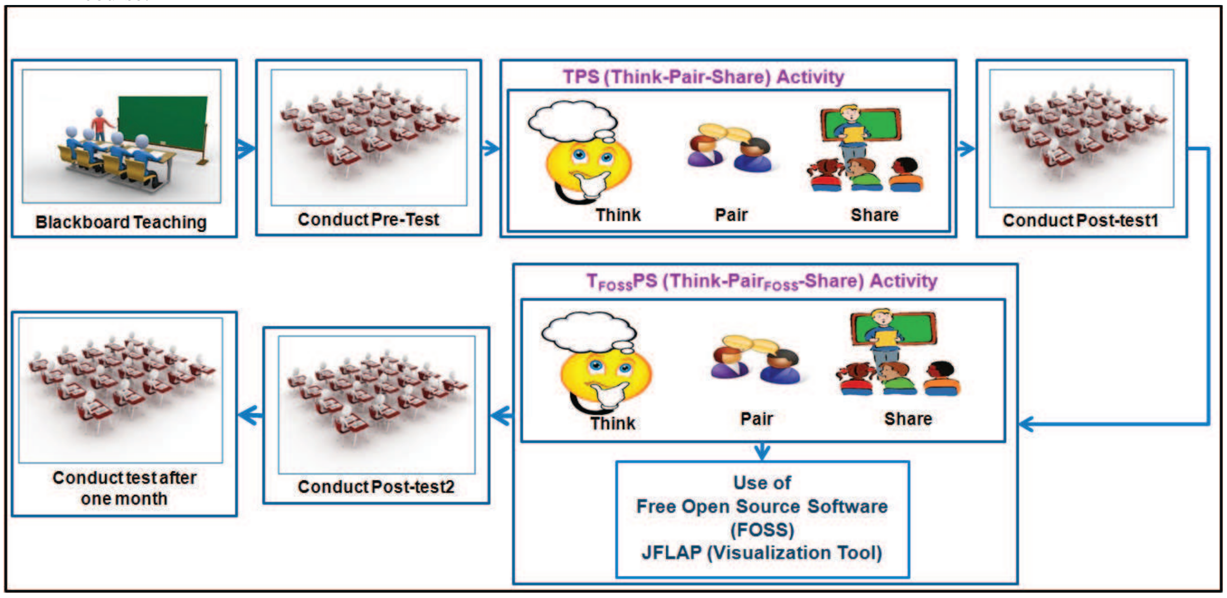

Blackboard Teaching
Conduct Pre-Test
TPS (Think-Pair-Share) Activity
Think
Pair
Share
Conduct Post-test1
T$_{FOSS}$PS (Think-Pair$_{FOSS}$-Share) Activity
Think
Pair
Share
Use of
Free Open Source Software
(FOSS)
JFLAP (Visualization Tool)
Conduct Post-test2
Conduct test after one month

Fig.4: Organization of Case Study

D.Pre-Test and Post-Test Questions

Pre-Test and Post-Test were conducted on the topic: Simplified Forms and Normal Forms. The weightage of all the tests was 30 marks. All the tests consist of questions like eliminating null productions, eliminate unit productions, eliminate useless variable and convert the given grammar into Chomsky Normal Form. These question covers Apply and Analyze level of Bloom's Taxonomy. The sample question in tests is shown as below:

```
Convert the following context free grammar to Chomsky
normal form
S->PQP
P-> 0P|^
Q-> 1P|^
```

E. Feedback

To understand student's perception, at the end of the activity, the feedback was taken on the various questions as mentioned in Table 1. Table 1 shows the feedback of TPS activity. From Table 1, 100% students agreed that TPS is useful activity.

TABLE.1: FEEDBACK FORM FOR TPS ACTIVITY

Sr. No.		**Never**	**Sometimes**	**Often**	**Always**
1	How frequently did you write the solution to the problem given by the instructor during the think phase?	2%	21%	27%	50%
2	How frequently did you discuss your solution with your partner during the pair phase?	0%	21%	21%	57%
		Strongly Disagree	**Disagree**	**Agree**	**Strongly Agree**
3	I stayed interested in the content of the lecture because of the think-pair-share activities.	0%	2%	70%	29%
4	Thinking about the problem and writing the solution during the think phase helped me learn <topic> concepts.	0%	0%	64%	36%
5	Discussing my solution with my partner during the pair phase helped me learn <topic> concepts	0%	4%	48%	48%
6	Listening to other students' solutions and discussion during the share phase helped me learn <topic> concepts.	4%	4%	57%	36%
7	I would not have learned as much from the lecture if there had been no think-pair-share Scale activities.	2%	30%	55%	13%
8	Did you like the Think-Pair–Share activity: Yes/No Why?	**TPS Yes= 100%**			

TABLE.2: FEEDBACK FORM FOR $TP_{FOSS}S$ ACTIVITY

Sr. No.		**Never**	**Sometimes**	**Often**	**Always**
1	How frequently did you write the solution to the problem given by the instructor during the think phase?	4%	10%	18%	68%
2	How frequently did you discuss your solution with your partner during the pair phase?	2%	14%	24%	59%
		Strongly Disagree	**Disagree**	**Agree**	**Strongly Agree**
3	I stayed interested in the content of the lecture because of the visualization tool: Parsing Simulator.	4%	2%	67%	27%
4	Thinking about the problem and writing the solution during the think phase helped me learn <topic> concepts.	0%	4%	53%	43%
5	Discussing my solution with my partner during the pair phase helped me learn <topic> concepts	0%	4%	61%	35%
6	Listening to other students' solutions and discussion during the share phase helped me learn <topic> concepts.	0%	14%	65%	20%
7	I would not have learned as much from the lecture if there had been no visualization.	4%	31%	53%	12%
8	Which one did you like the most and why a)TPS b)$TP_{FOSS}S$	**TPS = 18%** **$TP_{FOSS}S$= 82%**			

Table 2 shows the feedback for $TP_{FOSS}S$ activity. From table 2, it is observed that 82% students like this modified TPS i.e. $TP_{FOSS}S$ activity due to its effectiveness.

IV. EXPERIMENTAL RESULT

Figure 5 shows the performance of students in pre-test and post-test for TPS activity in TOC. The figure shows the significant improvement in students' performance in post-test.

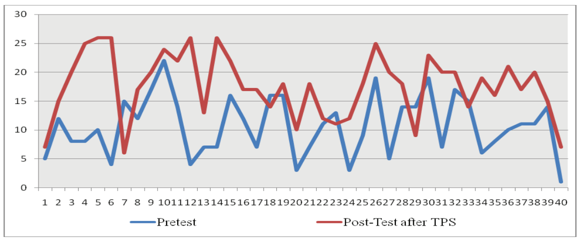


Fig.5: Performance of Students in pre-test and post-test 1 for TPS activity in TOC

Figure 6 compares the performance of students in pre-test and post-test1 for TPS activity and post test 1 and post-test2 for TP$_{FOSS}$S activity in TOC while Figure 7 shows performance of Students in pre-test, post-test1, post-test2 and test after one month in TOC

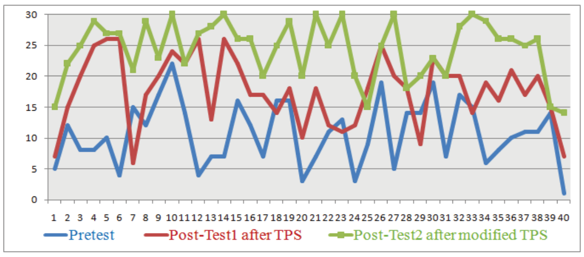


Fig. 6: Performance of Students in pre-test, post-test 1 for TPS activity & post test 1 and post-test2 for TPFOSSS activity in TOC

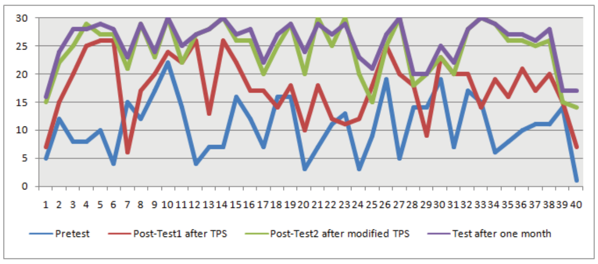


Fig. 7: Performance of Students in pre-test, post-test1, post-test2 and test after one month in TOC

Table 3 shows the pre-test level, number of students, mean of pre-test, mean of post-test 1 and mean of post-test 2 while figure 8 shows the graph for the same.

TABLE 3 : MEAN OF PRE-TEST, POST-TEST 1 AND POST-TEST2 IN TOC

Pre-Test Level	No of Students	Mean of Pre-Test	Mean of Post-Test1	Mean of Post-Test2
High (>=60 %)	3	20	24	26
Medium (40-59%)	15	14	16	24.2
Low (<40%)	22	6.9	18	24.2

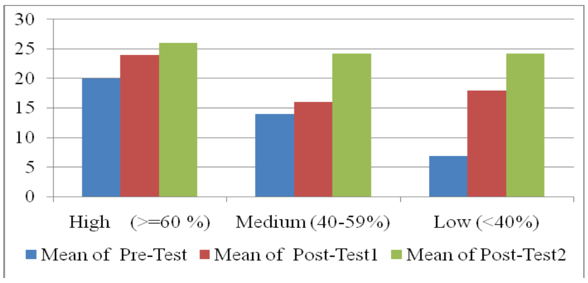


Fig. 8: Graph for Mean of pre-test, post-test 1 and post-test2 in TOC

t-Test is used to determine if two sets of data differ significantly from each other. For t-Test to be significant statistically, t value must be 2.145 and p value must be less than or equal to 0.05. t-Test result for pre-test and post-test1 is shown in table 4 while t-Test result for post-test1 and post-test2 is shown in table 5. t-Test result also shows statistical significant difference between pre-test and post-test conducted for this activity

TABLE 4 : T-TEST RESULT FOR PRE-TEST AND POST-TEST1

Degree of Freedom	Standard Deviation	t value	p value
78	5.33	-5.81	<0.0001

TABLE 5 : T-TEST RESULT FOR PRE-TEST1 AND POST-TEST2

Degree of Freedom	Standard Deviation	t value	p value
78	5.17	-5.84	<0.0001

V. CONCLUSION

In this paper, how TPS and TP$_{FOSS}$S activities can be used effectively for the course Theory of Computation is explained. In TP$_{FOSS}$Sactivity, as free open source tool is used which is GUI based, therefore teaching-learning is made more effective. Theory of Computation course is difficult to understand from student's perspective, which can be overcome by using a JFLAP tool with TPS which demonstrates all steps for the solution of example. Students will develop It is possible for students to practice more examples using FOSS. This tool helps the students at the time of examination to solve extra problems. Instructor's time in TPS activity will be saved as solution of problem will be visible on FOSS. This method can also be extended for teaching other engineering courses.

REFERENCES

[1] Mukta Goyal, Shelly Sachdeva "Enhancing Theory of Computation Teaching through Integration with other Courses", International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009

[2] Carlos I. Ches˜nevar et al. "Didactic Strategies for Promoting Significant Learning in Formal Languages and Automata Theory", ACM 1581138369, ITiCSE'04, June 28–30, 2004.
[3] Carlos Iv´an Ches˜nevar et al. "Teaching Fundamentals of Computing Theory: A Constructivist Approach", JCS&T Vol. 4 No. 2, August 2004.
[4] Carlos I. Ches˜nevar et al. "Using Theorotical Computer Simulators for Formal Laguages and Automata Theory"
[5] Anna O. Bilska et al."A Collection of Tools for Making Automata Theory and Formal Languages Come Alive"
[6] Aditi Kothiyal et. Al,"Effect of Think-Pair-Share in a Large CS1 Class: 83% Sustained Engagement", ICER'13, August 12–14, 2013.
[7] Gargi Banerjee et.al "Teaching with visualizations in classroom setting: Mapping Instructional Strategies to Instructional Objectives", IEEE Fifth International Conference on Technology for Education, 2013
[8] Sunita B. Aher, Dattatray P Gandhmal "$\text{TP}_{\text{FOSS}}\text{S}$: A Modified TPS Technique to Improve Student's Conceptual understanding of Compiler Construction Course", selected in T4E-2014
[9] http://en.wikipedia.org/wiki/Free_and_open-source_software
[10] http://www.jflap.org/