A learnable-by-design (LEAD) model for designing experiments for computer science labs

Mrityunjay Kumar
IIIT Hyderabad
Hyderabad, India
mrityunjay.k@research.iiit.ac.in

Venkatesh Choppella
IIIT Hyderabad
Hyderabad, India
venkatesh.choppella@iiit.ac.in

Sanjana Sunil
IIIT Hyderabad
Hyderabad, India
sanjana.sunil@students.iiit.ac.in

Sumaid Ali Syed
IIIT Hyderabad
Hyderabad, India
sumaid.ali@students.iiit.ac.in

***Abstract*—India graduates 1.5 million engineering students every year, majority of them from its Tier 2 and Tier 3 colleges where there is severe shortage of qualified faculty and lab infrastructure, and where English is second or third language for most students. To provide affordable virtual laboratories to these engineering colleges, Government of India runs the Virtual Labs project with a select set of participating institutions to create virtual labs required for entire engineering student population of India. The program suffers from two problems: 1) lack of focus on pedagogy and learnability of labs being created, and 2) little ability to scale to more teachers and subject matter experts to produce labs aligned to local contexts. This paper presents a model (LEAD model) for the design of virtual labs for a data structures and algorithms course in computer science and is intended to address the problems stated above. The model encodes the principles of learning in the design and structure of an experiment so that labs can be learnable by design. The model is based on constructivist theory of learning. It uses Bloom's Taxonomy for defining learning objectives and applies Merrill's first principles and Gagne's 9 events of instruction as instructional design methodologies. To ensure tasks in an experiment indeed aid learning, the model requires a mapping between tasks and learning principles to be created and published along with the experiment. The model uses a pedagogy that lays strong emphasis on conceptual understanding without the use of code or program. To demonstrate the efficacy of this model, we present the data analysis of the feedback from early users of one the experiments. Even though limited in scope, data shows that experiments built using LEAD model can aid understanding for the students.**

***Keywords*—virtual labs, undergraduate engineering, data structures, bloom's taxonomy, Gagne's 9 events of instruction**

I. Introduction

A virtual lab provides hands-on experience to engineering students and augments their understanding of the concept taught in the classroom. It is a learning resource that the student engages with, with or without a teacher's supervision. The utility of virtual labs in science education is well-established [1].

Of the 1.5 million engineering students graduating every year, hardly 6-12% of these students are employable. While employability is not a great measure of learning, the statistic certainly indicates lack of learning. To address this challenge, The Ministry of Human Resource Development, India initiated Virtual Labs - a large-scale open source collaborative project for creating simulation-based engineering labs that can be accessed remotely over the internet [2]. Target audiences include, among others, students from engineering colleges from Tier-2 and Tier-3 cities who lack access to physical lab infrastructure and good faculty. They also don't have English as their primary language. Teachers frequently use local language and context while teaching these students.

One of the authors (Kumar [3]) reviewed the usability of some of these existing Virtual Labs from technical as well as pedagogical usability perspective and found the labs to be lacking in many of the usability dimensions. One of the recommendations in the paper was to define a consistent model of the lab so that design and implementation inconsistencies can be reduced. This paper presents a model of an experiment that addresses this recommendation.

Given the diversity of language, culture and resource access across colleges in India, there is a need to involve local teachers and subject matter experts to produce these labs that are contextualized to local needs. However, involving them requires addressing two challenges:

- Labs may be inconsistent and not aligned to learning principles.
- Teachers may not have the capability to produce these labs

The model we have built takes these into account when proposing a design of the experiment, ensuring consistency and adherence to the learning principles, while allowing local teachers and subject matter experts to design these experiments. This model is influenced by multiple learning and instructional design theories that apply to laboratories and online learning, particularly three key ones:

1) Constructivist learning theory (for specifying the elements of structure of the experiment)
2) Bloom's taxonomy (for specifying the learning objectives)
3) Merrill's first principles and Gagne's 9 events of instruction (for specifying instructional design principles to be applied)

In the rest of the paper, we refer to this model as learnable-by-design (LEAD) model, Fig. 1 describes the model visually.

When defining LEAD model, we focused on 3 key attributes:

978-1-7281-4227-2/19/$31.00 ©2019 IEEE
DOI 10.1109/T4E.2019.00050

1) Define an abstract model that is strongly driven by learning principles and pedagogy so that learnability can be designed into the model
2) Define a concrete structure of the experiment that encodes the learning principles so that learnability doesn't get lost during construction phase
3) Define a mapping between learning principles and structural elements so that learnability can always be validated

To ascertain the utility of such a model, primary research question we posed was this:

RQ1 Does an experiment created using LEAD model help students understand the topic better?

RQ2 Is it easy to create an experiment using LEAD model?

To answer RQ1, we asked a small group of first year undergraduate computer science students to use this "Stacks and Queues" experiment and collected their response to the questions about how well this aided their understanding of the topic.

To answer RQ2, the paper records our observations from having a team of 2nd year undergraduate students create one of the experiments using this model.

The experiment we use, 'Stacks and Queues', is part of a series of (twenty-three) experiments being built for two Data Structures and Algorithms labs, and a subsequent paper will report the results from these other experiments.

Rest of the paper is organized as follows. Section 2 discusses related work, from model as well as implementation perspective. Section 3 presents the experiment model and the motivation behind the structure. Section 4 presents the experiment implementation (stacks and queues) and its mapping to learning principles and model structure. Section 5 presents the user feedback and its insights. Section 6 concludes the paper and points to the future work.

II. Related work

We looked at the literature around how various learning theories and principles have been applied in the context of virtual experiments and labs.

Technology innovations like MOOC have altered the landscape of learning [4]. However, there are concerns on the pedagogy being used, esp. lack of focus on labs for science education [5]. Laboratories aid in conceptual understanding of theoretical concepts via hands-on experience, and virtual labs bring value to students [1]. Effectiveness of the virtual labs has been questioned sometimes but multiple studies reiterate the utility of virtual labs [6] [7]. New technologies are fueling the growth of virtual laboratories [8]. These reiterate the fact that Virtual Labs initiative is the right way forward for a country like India for furthering their engineering education.

For Virtual labs, visualization is an important tool for teaching concepts. However, lack of widespread usage and effectiveness challenges around visualizations [9] [10] [11] is concerning. As Hundhausen suggests in [12], 'Despite the intuitive appeal of the technology (visualization), it has failed to catch on in mainstream computer science education', but he does conclude that 'algorithm visualizations are educationally effective insofar as they enable students to construct their own understandings of algorithms through a process of active learning'. Hansen suggests in [13] that interactive content or visualization is not enough, a focus on pedagogy is essential and yields results when applied.

A good lab or experiment needs to apply sound learning theories and principles. Constructivist theory of learning has been the most prevalent approach towards science learning [14] [15]. Its implication for instructional practices and labs has also been well-studied [16]. We find it suitable for virtual labs where focus is on hands-on activities and interactions for practice and exercises. Fuller [17] presents Bloom's taxonomy [18] as the most cited in the literature for taxonomy. Manaris [19] reports using Bloom's taxonomy for defining learning objectives and finds it a useful tool for defining and refining objectives as well as for assessment. Starr [20] applies them to define assessable learning objectives throughout the CS curriculum.

Ertmer [21] discusses elements of instructional design for Constructivist learning theory (anchoring, active use of learning, multiple presentations, use of problem-solving skills). Merrill's first principles [22] and Gagne's 9 events of instructions [23] [24] are the most prevalent instructional design principles aligned to constructivist theory.

Technical and pedagogical usability [3] are key consideration for virtual labs. So is a clearly defined structure [25]. LEAD model includes these elements, though some of these aspects will be more relevant during construction phase of the experiment. The survey of related work suggests that virtual labs require a strong pedagogy, visualization is useful for engagement but can't be used alone, and that constructivist theory, bloom's taxonomy and Merrill's and Gagne's instructional design principles, are very aligned to virtual labs.

A. No interactive content in MOOCs

We also did a short analysis of MOOC courses and their use of labs or other interactions (results shared in Section III), given their widespread use in learning computer science topics. Results continue to be disappointing - there is very little evidence of hands-on, interactive elements or experiments being used.

We reviewed data structures and algorithm courses on 4 MOOCs. Table 1 summarizes the courses we reviewed for each of these MOOC. We picked distinct courses, some of them had similar courses - for ex, given 2 courses, 'data structures in python' and data structures in C', we picked only one. When searching, we used keywords 'Data structures', and 'Algorithms'.

All the courses had similar structure - lectures and quizzes. There was no interactive content used in any of the courses. One course had java applet which didn't work on our browser, NPTEL had a few simulations but they required Windows 32-bit, so again couldn't be run. Clearly, the focus of these courses has not been on virtual experiments and interactive content.

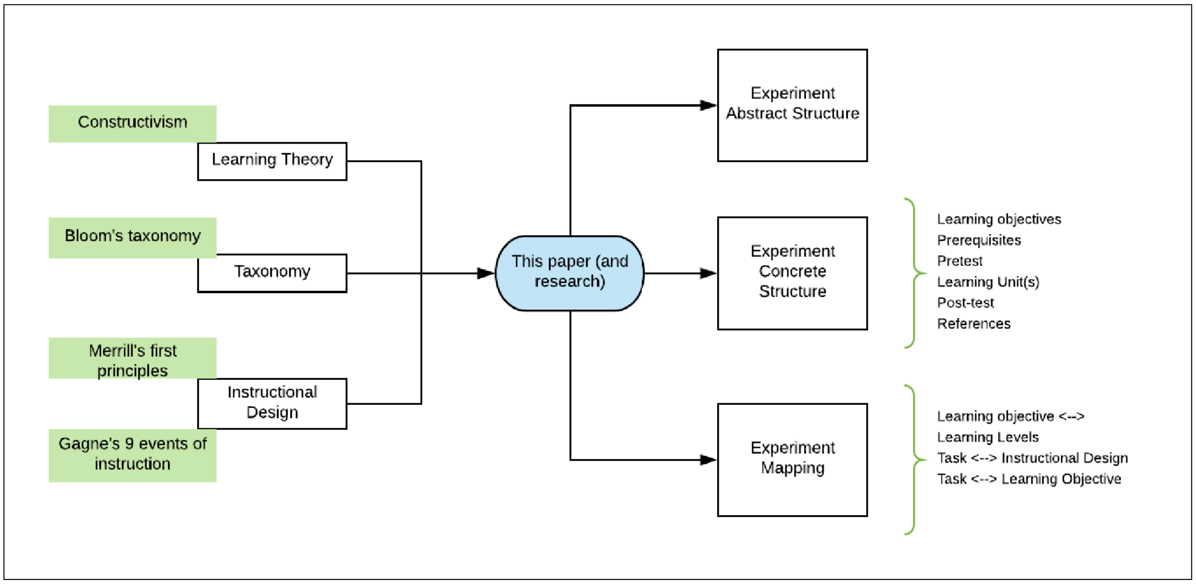

Fig. 1: Elements of LEAD model

TABLE I: MOOCs evaluated

MOOC	Courses
edX	Algorithms and Data Structures Data Structures Fundamentals Data Structures: An Active Learning Approach Data Structures and Software Design:Getting Started Python Data Structures
Udemy	Python for Data Structures,Algorithms.. Practical Data Structures and Algorithms.. The Coding Interview Bootcamp: Algorithms + .. Data Structures and Algorithms Through C ..
Coursera	Data Structures and Algorithms Specialization Algorithms Specialization
NPTEL	Computer Algorithms 2 Data structures and algorithms Design and analysis of algorithms Programming, Data structures and Algorithms

III. Experiment Model and Structure

In LEAD model, an experiment consists of one or more Learning Units and each learning unit consists of tasks that help to achieve a learning objective and use one or more instructional design principles to achieve these objectives.

A typical experiment structure for a LEAD model looks like this:

- Learning objectives
- Prerequisites
- Pretest
- Learning Unit(s)
  - Task 1 (Introduction): Read the concept
  - Task 2 (Demo): See the demonstration of the concept
  - Task 3 (Practice): Practice using interactive elements
  - Task 4 (Exercise): Check understanding of the concept
- Post-test
- References

To ensure accessibility to a wider audience, each learning unit also includes a video presentation of the entire unit so that learners who prefer lecture-like setup can also benefit.

An experiment consists of a set of modules. A learning unit is a module, so is a pretest or post-test. Next section provides more details of each of the module of the structure. In a forthcoming paper, we will describe the formal structure and ontology of a virtual experiment that aligns to this model.

To ensure that these learning principles are adhered to as the experiment is designed, constructed and released, the model defines three mappings that an experiment creator must produce:

1) Learning objective mapped to Learning Levels
2) Tasks mapped to Instructional Design principles
3) Tasks mapped to Learning Objectives

These mappings are made available as part of the experiment and are available to validate any of the intermediate steps an experiment goes through. The details of the mapping and its usage during construction process will be reported in a forthcoming paper. The focus of this paper is to describe the model and illustrate it with an experiment implementation.

A good design (of a learning content) focuses on four aspects:

1) Learning Theory
2) Learning Taxonomy
3) Instructional Design
4) Pedagogy

A. Learning Theory

A model should implement a learning theory that has strong alignment to the learning needs of the target segment. Given the unique applicability of constructivist theory to laboratory learning and its widespread use in science education [26], this is the theory LEAD model has adopted. Constructivist theory suggests that all knowledge is constructed, and all learning is a process of construction [27]. This is closely aligned to the objectives of virtual labs - allow students to learn on their own pace, let them actively engage with the content and allow hands-on experiences. LEAD model defines a Learning Unit to be composed of four predefined task types (Introduction, Demonstration, Practice, Exercise), thereby aligning with the laboratory process and constructivist approach.

B. Learning Taxonomy

Pedagogical philosophy for the LEAD model is influenced by Bloom's taxonomy. The model maps the learning objectives of the experiments to learning levels in Bloom's taxonomy to ensure that all objectives are aligned to the learning levels required by the experiment and the creator. The learning objectives section of the experiment structure will list course level and topic level learning objectives, using a specific way to write the objectives (Fig. 2) that is based on Gagne's recommendation on writing objectives. Each objective will also specify which learning level of Bloom's taxonomy does this map to. Experiments are expected to have objectives for 'remembering', 'understanding' and 'applying' (first 3) levels at least. Using the mapping between task and learning objective, it is possible to identify whether there are tasks to achieve any given learning objective.

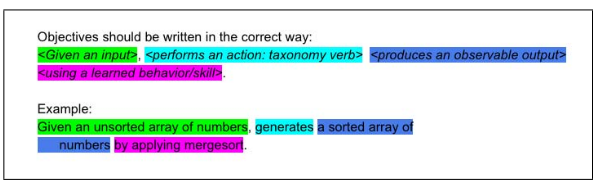

Fig. 2: Writing objectives (Gagne's style)

C. Instructional Design

A well-designed content must use instructional design methodologies that are aligned to the learning theory they are based on. Merrill's first principles and Gagne's 9 events of instructions are primary instructional design strategies in LEAD model. We use Merrill's first principles because it is more foundational in nature and well-connected to constructivism learning theory. We also use Gagne's nine events of instruction as it serves as the elaboration of Merrill's first principles and makes the motivation of instruction more explicit. Using the mapping that each experiment provides between task and instructional design principles; it is possible to identify whether a task is following right principles or not.

D. Pedagogy

Our pedagogy for teaching these data structures and algorithm experiments focuses on providing a conceptual understanding of the topics. We apply a pedagogy that focuses on visual representations and focuses more on building concepts rather than teaching programming of these data structures and algorithms. We place strong emphasis on visualization and interactivity. We also avoid using simulations and interactive environment alone and instead focus on using interactive artifacts in specific context only (Practice and Exercise tasks, sometimes in Demo tasks that can use light interactivity).

IV. Experiment Implementation

This section describes the elements of an experiment that we built using this model, Stacks and Queues. However, it is important to note that what we show in these subsections are really experiment user's view of the model - experiment creator will provide different (mostly more) content which gets transformed through construction process into user's view of the experiment.

Each element of the structure was implemented keeping in mind the principles of LEAD model. Our key focus was to ensure we are applying good instructional design principles. We achieved this by mapping our tasks to Gagne's 9 events of instructions:

1) Gain attention
2) Inform learners of the objective
3) Stimulate recall of prior learning
4) Present the stimulus
5) Provide learning guidance
6) Elicit performance
7) Provide feedback
8) Assess performance
9) Enhance retention and transfer

We also used the usability checklist provided in Kumar [3] to do usability review of the content. Some aspects of the checklists were already implemented in the framework that we used to build these experiments.

A. Learning Objectives

Before starting, the learner is presented with a set of statements that informs them of the skills they would expect to gain by completing the experiment. In the stacks and queues experiment, the objectives were listed out as follows:

- Gain a basic understanding of stacks as an abstract data type.
- Understand stacks operations and associated time complexities through interactive animations
- Understand applications of stacks

Taxonomy Mapping - First two objectives map to second level of Bloom's Taxonomy: 'Understanding', while last one maps to third level, 'Applying'.

The LEAD model recommends a structure for constructing learning objectives, however that was not used when presenting it to the user to make it easier for users to comprehend

Instructional Design Mapping - Gagne's second event of instruction: 'Inform learners of the objective'

B. Prerequisites

The learner is presented with the prerequisites for learning the content of that experiment so that he or she does not face any difficulty later. In our experiment, the prerequisites were listed out as follows:

- Basic knowledge of arrays and linked lists
- Basic understanding of time complexity notations

Resources were provided for readers who wanted to brush up their knowledge.

Instructional Design Mapping - Gagne's third event of instruction: 'Stimulate recall of prior learning'. This is a weak mapping because there is no stimulation, only recall via reference to resources for these prerequisites.

C. Pretest

The pretest section consists of multiple-choice questions based on the topics listed out in the prerequisites module of the experiment to establish a subject knowledge baseline. After the student completes the pretest, their score along with the correct answers are displayed on the screen. In addition to the score, the student is also provided links to sources to learn the prerequisite topics.

Instructional Design Mapping - Gagne's third event of instruction: 'Stimulate recall of prior learning'. The questions are picked to be stimulating and not only answered by rote learning. This module also falls under the Activation Phase described by Merrill.

Next few sections are repeated for each learning unit that the experiment contains.

D. Introduction

The student is given a gentle introduction to the topic by drawing an analogy with a real-life object or situation. The topic is then formally introduced by giving a simple definition or a concise explanation. While designing the experiment, more focus was given on visualization than on text in order to enhance knowledge transfer and help the student retain information longer [28].

In the stacks and queues experiment, the topic was introduced by drawing an analogy to a stack of books and giving a short description of push and pop operations. Images of the operations were also given. A few applications of stacks were given as well, for example, undoing changes in a text editor and reversing a word.

Instructional Design Mapping - Gagne's fourth event of instruction: 'Present the stimulus'. The introduction used engaging material, including visual elements, to stimulate the interest of the student.

E. Demonstration

After the introduction, the user is given interactive demos to help them understand and visualize basic concepts better. This individual practice activity builds comfort with the topic [29]. Students remember better when teaching of a topic involves demonstrations [22]. In the stacks experiment, the user was presented with an empty representation of a stack with an option for the user to enter a value and push the value with the press of a button and pop the value with the push of another button (Fig. 3). There were also separate interactive demos for stacks represented using an array and stacks represented using a linked list.

Instructional Design Mapping - Gagne's fifth event of instruction: 'Provide learning guidance'. The demonstration is designed such that user can understand the steps involved in an operation which helps the learning process.

F. Practice

The student is provided with interactivities that allow the student to engage with it and practice what they have learned through introduction and demonstration. This is where they put their learning into use. Practice provides rich feedback to the user so that they can understand and validate their learning. In this experiment, we merged Practice with Exercise.

Instructional Design Mapping - Gagne's seventh event of instruction: 'Provide feedback'. The practice provides rich feedback for every interaction (if appropriate) and helps the students understand the gaps in their understanding and bridge them.

G. Exercise

The students' conceptual understanding is tested using interactive exercises. The student is presented with an interactive problem and allowed to interact with it to solve the problem. This proceeds similar to Practice, but provides very little feedback during the interaction, and provides the feedback only at the end of the activity.

In one artefact of this experiment, we designed an activity, asking the user to sort an array of numbers using two interactive stacks, given primitive operations of push and pop.

This experiment did not include Practice (one of the recommended tasks) explicitly, it was merged with Exercise.

Instructional Design Mapping - Gagne's fifth event of instruction: 'Provide learning guidance', as well as sixth event of instruction: 'Elicit performance'. The exercise helps in better learning, and at the same time evaluates the learner and make them perform.

H. Post-test

The post-test is the final test of the concepts learned by students from the experiment. Like the pretest, the students receive feedback on their performance, along with the correct answers of the quiz.

The experiment on stacks asked, as an example, 'What is the time complexity of finding the minimum element of the stack?'. The questions had multiple-choice answers and were designed to check the understanding of each learning unit.

Instructional Design Mapping - Gagne's eighth event of instruction: 'Assess performance'. The post-test is designed to evaluate student's learning and assess the performance.

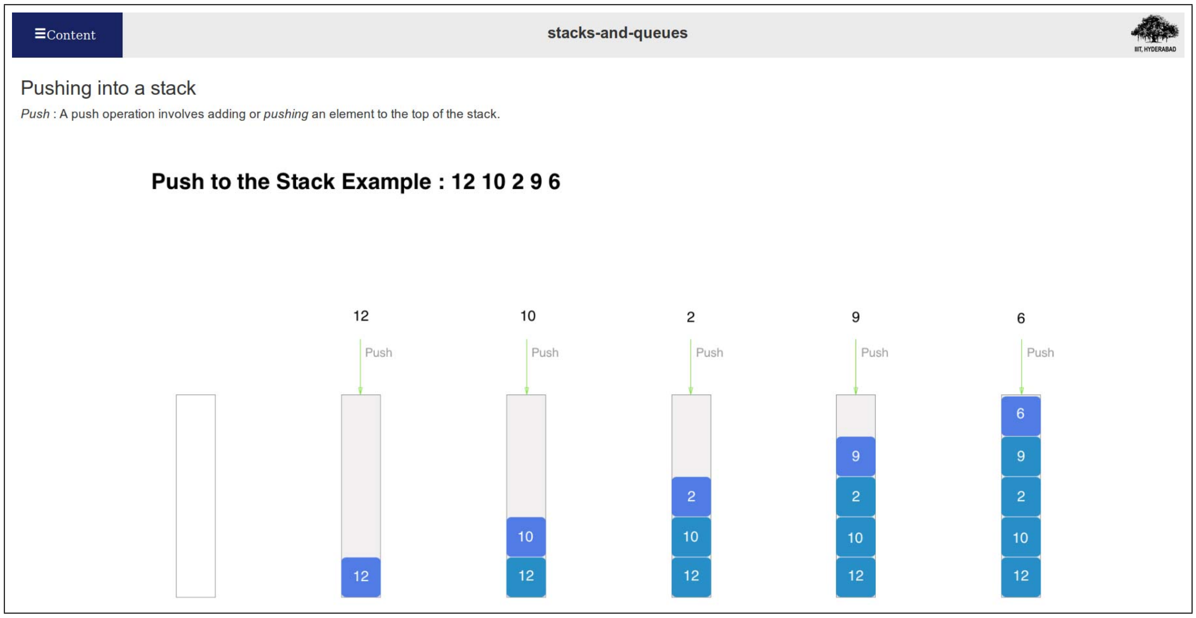

Fig. 3: Stack Push Demonstration

I. References

This module consists of links and references that discuss the topic further and provides more detail. This is useful for students who want to learn more, and it also serves the purpose of connecting other real-life problems with what the student has learned.

Instructional Design Mapping - Gagne's ninth event of instruction: 'Enhance retention and transfer'. This is a weak mapping. References are selected to extend the knowledge gained by referring to additional material as well as advanced applications of the subject.

V. User Feedback

41 students were instructed to try an experiment and fill in an online questionnaire. The group primarily consisted of first-year Computer Science students who had some basic knowledge of computer science and were familiar with the prerequisites of the experiment. We looked at the descriptive statistics for the data, given it was a small sample size.

A. Data Analysis

Gathering data from students (and analyzing them) is one of the key ways to validate this model. This section presents results of analyzing user feedback to this experiment. We will continue the work on surveying a larger sample and with more experiments as they get built.

Our focus was to primarily get feedback on whether the experiment helped the students understand the concept better. We checked for understanding by asking 3 questions:

- Understanding - How much did the experiment help you understand the topic?
- Engagement - How engaging was the experiment?
- Interactive elements - How helpful were interactive elements in understanding content?

To understand the prior knowledge of students about the topic, we included the question "How much of the topic you knew before?"

Fig. 4 presents the mean (M) and coefficient of variation (CV) for each of the questions. All student responses lie between 0 and 10.

We also wanted to ensure that the 3 questions we asked measure the same basic point we wanted to address: does the experiment help them understand better? So we also ran a Principal Component Analysis (PCA) to understand the underlying factors. See Fig. 5 for details. We notice that only 2 components (PC1 and PC2) can explain almost 90% of the data. PC1 is primarily composed of understanding questions, and PC2 almost completely explains prior knowledge.

From Fig. 4, we notice there is high mean for the understanding-related questions (7.7-8.2) and low CV (0.20-0.23). We conclude that the students found the experiment helpful in understanding the topic better.

We noticed quite some variation in prior understanding of the topic for students (Mean: 6.2, Median: 7.5, CV: 0.53). High median suggests many students had excellent prior knowledge, while low mean suggests some students had very little knowledge. This is interesting because this suggests that even a diverse background of the sample found the experiment aids their understanding.

Normally, we would want students to have good prior exposure to the topic (since assumption is students have done the topic in classroom setting). However, given the characteristics of the target segment (lack of teachers and resources), we need to test the efficacy of the experiments under varied prior

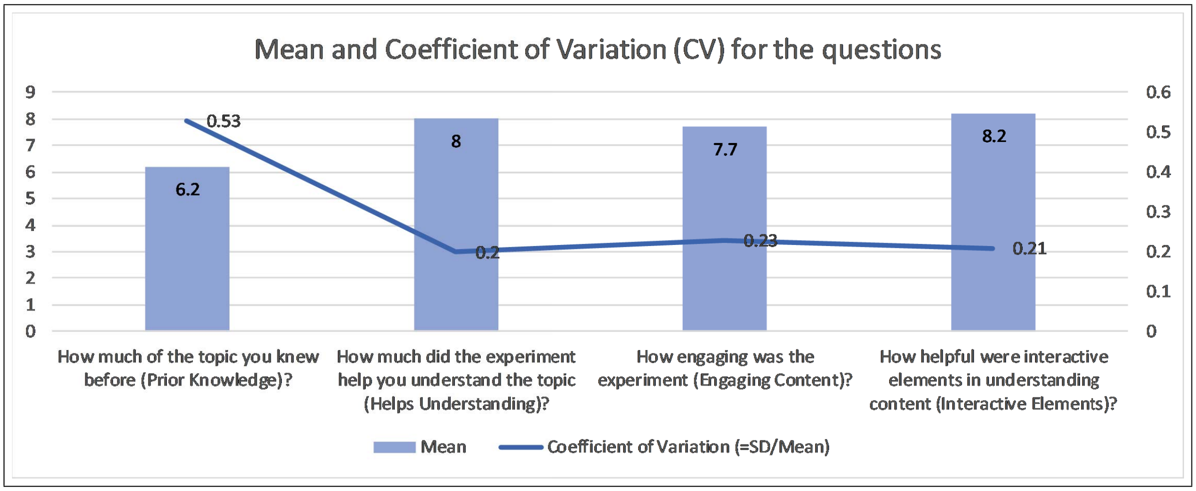

Fig. 4: Questions - Mean and Coefficient of Variation

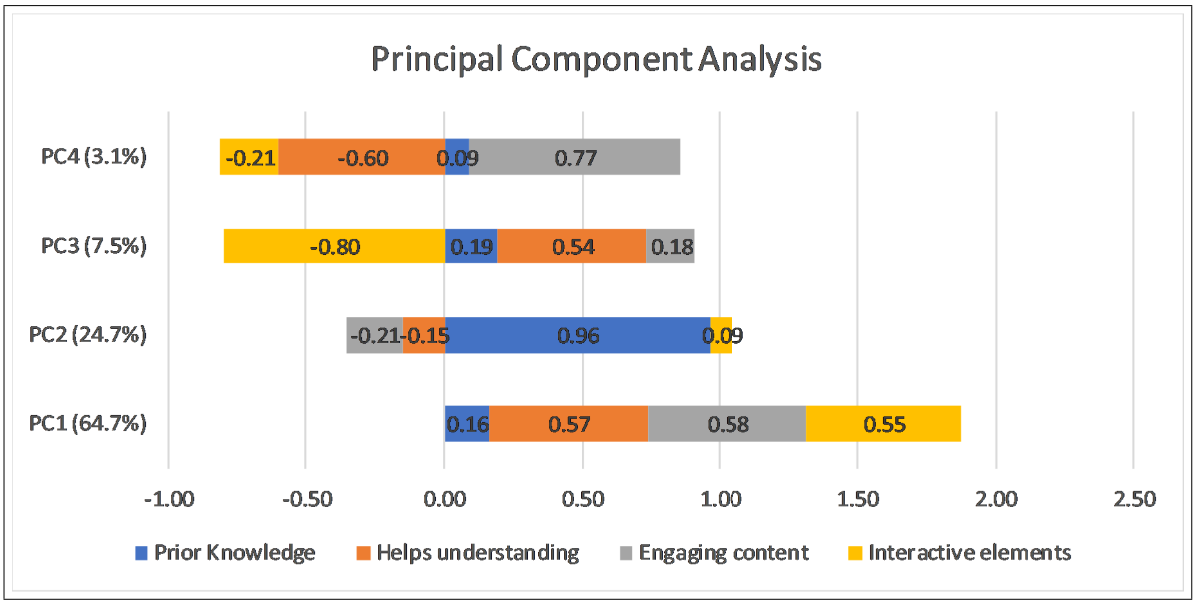

Fig. 5: Questions- PCA

knowledge and ensure they remain effective even with low prior knowledge. This will be a topic for further research.

VI. Conclusion and Future Work

This paper presented the LEAD model for creating experiment structure. This model is aligned to constructivist learning theory, learning levels of Bloom's taxonomy, and Instruction Design principles of Merrill's and Gagne's. The LEAD model seeks to address the problem of engineering students in Tier 2 and Tier 3 cities (lack of good faculty and resources). We used the details of an experiment we built, to illustrate the LEAD model structure. We also analyzed the user feedback for this experiment. The data suggests that the experiment aided the understanding of the topic for the students.

We had posed 2 research questions:

RQ1 Does an experiment created using LEAD model help students understand the topic better?

RQ2 Is it easy to create an experiment using LEAD model?

We conclude that the experiment created using LEAD model does help the students understand the topic better. We also conclude that it is easy to create an experiment using LEAD model even with average skill set (given the use of 2nd year undergraduate students for creating the experiement).

There is a need to expand the user testing cycle and do it with large sample size of students and many more experiments so that generic inferences about the model can be drawn. Future work should also be done in provide additional theoretical basis for why such a model will work to produce learning by design, including comparison with similar other models for designing experiments.

References

[1] J. R. Brinson, "Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research," *Computers & Education*, vol. 87, no. Supplement C, pp. 218 – 237, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131515300087

[2] R. Bose, "Virtual labs project: A paradigm shift in internet-based remote experimentation," *IEEE access*, vol. 1, pp. 718–725, 2013.

[3] M. Kumar, J. Emory, and V. Choppella, "Usability analysis of virtual labs," in *2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)*, July 2018, pp. 238–240.

[4] L. Yuan, S. Powell, J. CETIS *et al.*, "Moocs and open education: Implications for higher education," 2013.

[5] M. M. Waldrop, "Education online: The virtual lab," *Nature News*, vol. 499, no. 7458, p. 268, 2013.

[6] J. E. Corter, J. V. Nickerson, S. K. Esche, C. Chassapis, S. Im, and J. Ma, "Constructing reality: A study of remote, hands-on, and simulated laboratories," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 14, no. 2, p. 7, 2007.

[7] T. A. Stuckey-Mickell and B. D. Stuckey-Danner, "Virtual labs in the online biology course: Student perceptions of effectiveness and usability," *MERLOT journal of online learning and teaching*, vol. 3, no. 2, pp. 105–111, 2007.

[8] A. Hofstein and V. N. Lunetta, "The laboratory in science education: Foundations for the twenty-first century," *Science education*, vol. 88, no. 1, pp. 28–54, 2004.

[9] C. A. Shaffer, M. Akbar, A. J. D. Alon, M. Stewart, and S. H. Edwards, "Getting algorithm visualizations into the classroom," in *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011, pp. 129–134.

[10] M. Knobelsdorf, E. Isohanni, and J. Tenenberg, "The reasons might be different: Why students and teachers do not use visualization tools," in *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*. ACM, 2012, pp. 1–10.

[11] E. Isohanni and H.-M. Järvinen, "Are visualization tools used in programming education?: by whom, how, why, and why not?" in *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*. ACM, 2014, pp. 35–40.

[12] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," *Journal of Visual Languages & Computing*, vol. 13, no. 3, pp. 259–290, 2002.

[13] S. R. Hansen and N. H. Narayanan, "Helping learners visualize and comprehend algorithms," *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, vol. 2, 01 2000.

[14] A. Colburn, "Constructivism: Science education's "grand unifying theory"," *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, vol. 74, no. 1, pp. 9–12, 2000.

[15] M. Ben-Ari, "Constructivism in computer science education," *Journal of Computers in Mathematics and Science Teaching*, vol. 20, no. 1, pp. 45–73, 2001.

[16] W. L. Saunders, "The constructivist perspective: Implications and teaching strategies for science," *School Science and Mathematics*, vol. 92, no. 3, pp. 136–141, 1992.

[17] U. Fuller, C. G. Johnson, T. Ahoniemi, D. Cukierman, I. Hernán-Losada, J. Jackova, E. Lahtinen, T. L. Lewis, D. M. Thompson, C. Riedesel *et al.*, "Developing a computer science-specific learning taxonomy," *ACM SIGCSE Bulletin*, vol. 39, no. 4, pp. 152–170, 2007.

[18] D. R. Krathwohl, "A revision of bloom's taxonomy: An overview," *Theory into practice*, vol. 41, no. 4, pp. 212–218, 2002.

[19] B. Manaris, M. Wainer, A. E. Kirkpatrick, R. H. Stalvey, C. Shannon, L. Leventhal, J. Barnes, J. Wright, J. B. Schafer, and D. Sanders, "Implementations of the cc' 01 human–computer interaction guidelines using bloom's taxonomy," *Computer Science Education*, vol. 17, no. 1, pp. 21–57, 2007.

[20] C. W. Starr, B. Manaris, and R. H. Stalvey, "Bloom's taxonomy revisited: specifying assessable learning objectives in computer science," in *ACM SIGCSE Bulletin*, vol. 40, no. 1. ACM, 2008, pp. 261–265.

[21] P. A. Ertmer and T. J. Newby, "Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective," *Performance improvement quarterly*, vol. 6, no. 4, pp. 50–72, 1993.

[22] M. D. Merrill, "First principles of instruction," *Educational technology research and development*, vol. 50, no. 3, pp. 43–59, 2002.

[23] R. M. Gagne, W. W. Wager, K. C. Golas, J. M. Keller, and J. D. Russell, "Principles of instructional design," *Performance Improvement*, vol. 44, no. 2, pp. 44–46, 2005.

[24] R. M. Gagne *et al.*, "The conditions of learning," 1965.

[25] M. Lister, "Trends in the design of e-learning and online learning," *Journal of Online Learning and Teaching*, vol. 10, no. 4, p. 671, 2014.

[26] R. Trumper, "The physics laboratory–a historical overview and future perspectives," *Science & Education*, vol. 12, no. 7, pp. 645–670, 2003.

[27] T. M. Duffy and D. J. Cunningham, "7. constructivism: Implications for the design and delivery of instruction," 1996.

[28] G. Shabiralyani, K. S. Hasan, N. Hamad, and N. Iqbal, "Impact of visual aids in enhancing the learning process case research: District dera ghazi khan." *Journal of Education and Practice*, vol. 6, no. 19, pp. 226–233, 2015.

[29] T. Wulf, "Constructivist approaches for teaching computer programming," in *Proceedings of the 6th conference on Information technology education*. ACM, 2005, pp. 245–248.