Development of mobile applications to study engineering students' patterns of learning

David S. Janzen
Computer Science and Software Engineering
California Polytechnic State University
San Luis Obispo, California, USA
djanzen@calpoly.edu

Ryan Chang
Education Research and Outreach
University of Oregon
Eugene, OR, USA
rchang@uoregon.edu

John Chen
Mechanical Engineering
California Polytechnic State University
San Luis Obispo, California, USA
jchen24@calpoly.edu

***Abstract*—This Research Work in Progress Paper presents a novel mobile software application created to perform ecological momentary assessment in two longitudinal studies with undergraduate engineering students. The studies collect learning activity data from first and second-year mechanical engineering students to examine the correlation of active learning with grit growth and student success and retention, and the possible associations between these factors. The software system consists of native Android and iOS mobile applications and a REST-based server application for prompting and collecting learning activity data from student participants. The system utilizes personalized student schedule data to push notifications to students on their smartphones as they complete each class or lab section. Students then submit the amount of time they spent in each class doing different types of activities including interactive, constructive, active, and passive. Students also receive notification reminders to enter out-of-class study activities. The system also includes automated daily audits and notifications to ensure consistent data entry activity. This paper will discuss the research context, early study results, software architecture, and the software development experience including institutional challenges, rationale for technology choices, strategies for developing research software with student developers, and lessons learned.**

***Index Terms*—Engineering Education, Application Software**

I. Introduction

Grit is a psychological trait defined as the passion and perseverance for long-term goals. Duckworth and co-workers presented evidence of correlations between grit and desirable academic outcomes including grade-point-average and, among West Point cadets, retention through the first year. [1]–[4]

The question is whether teachers can influence a student's degree of grit. A team of faculty at California Polytechnic State University (Cal Poly) designed a longitudinal semi-controlled experiment to investigate if active learning builds grit among students. A key element of this study was the in-the-moment collection of student data on the types of learning activities they experienced. Android and iOS apps along with a Parse server application were developed to accomplish this data collection.

This material is based upon work supported by the National Science Foundation under Grants No. (1431879, 1561623). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

This paper reports on the development of these applications which were given the name "Actively Learning" (referred to as ALApp). Section II describes the study context including a brief discussion of the research method. Section III reports some initial study findings. Section IV describes the technologies used in ALApp. Section V describes considerations for academic researchers who desire to utilize undergraduate computing students to develop similar applications. Section VI describes some institutional challenges encountered along the way, and Section VII summarizes a few lessons learned. As a Work in Progress paper, the authors seek feedback on their ideas of how the system could be generalized for use in a variety of education research studies.

II. Context

Two longitudinal semi-controlled experiments were designed to examine the influence of active learning on students' level of grit. All participants began as first-year Mechanical Engineering students at Cal Poly which has an enrollment of over 6,400 students in the College of Engineering with over 1,200 students in Mechanical Engineering. In Study A, first-year students were tasked with tracking their learning activities within all math, science and engineering courses through their first two years of study. In Study B, second-year students were tasked with tracking their learning as they progressed through the mechanics (applied physics) sequence, which spans roughly 4-6 academic terms, depending on when students choose to take certain courses or if they take an extended internship or study abroad. All participants were compensated monetarily and participation had no effect on any course grade. The studies are described in more detail in a separate publication within this conference [5].

A. Research Method

Studying engineering involves learning in a variety of contexts and activities. Students experience various levels of active learning through attending lectures, completing homework assignments, preparing for class, studying for quizzes and examinations, and seeking additional help. To describe these experiences for a complete measure of each student's type (or quality) and quantity of active learning, this project uses an educational framework developed by Chi and coworkers [6],

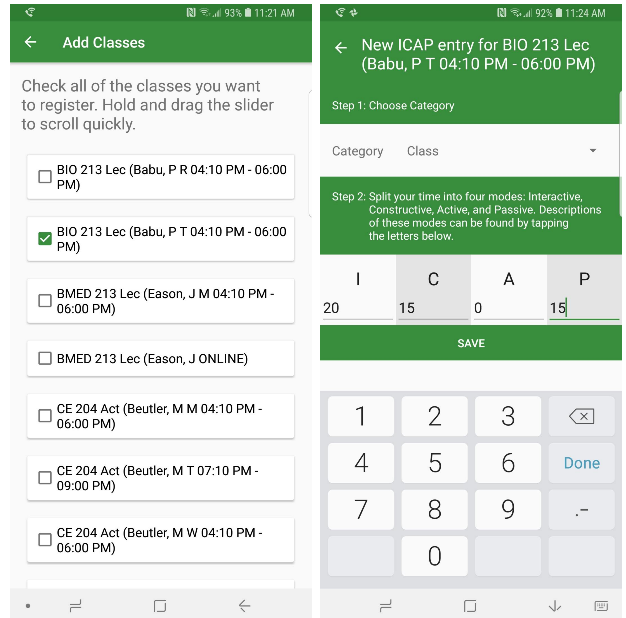
[7]: the interactive-constructive-active-passive (ICAP) differentiated learning activities.

The ICAP framework classifies learning activities by observable, overt actions of the learner. A *Passive* learning activity is one in which the learner essentially engages in no overt actions. Listening to a lecture, watching a video, and reading text are examples. By contrast, an *Active* learning activity is characterized by overt actions that demonstrate paying attention. Examples include note-taking or highlighting of text. (Note that at this point active learning has taken on a definition that is quite different than the general use of the term in education, which would classify note-taking as a "passive" learning activity.) If the learner goes one step further and generates additional knowledge or information beyond that which is provided, she is engaging in *Constructive* learning. Solving homework problems alone or resolving questions while reviewing notes alone are examples. The final category of *Interactive* learning requires learners to interact with someone (e.g., a peer or expert) or something (e.g., a computer tutor) in order to build on the provided information. There must be an exchange of information between the members, such as defending one's responses, responding to questions, or correcting noted errors. The conventionally accepted active-learning techniques [8]–[11] would be classified as either constructive or interactive in the ICAP framework. Furthermore, based on the possible underlying cognitive mechanisms being activated by each kind of activity, the expected learning gains should increase in the order of passive $<$ active $<$ constructive $<$ interactive; this is supported by the studies cited by [7].

This project adopts the ICAP framework to measure the "quality" of active learning experienced by students, with passive learning being lowest quality and interactive learning as the highest. The "quantity" of active learning is simply the amount of total time students expend in each of the ICAP categories for each course. Students will no doubt experience learning across this range of activities in a course, and across the curriculum as they traverse it with various instructors.

Study participants were given training on the ICAP framework to discern the differences between the levels of activity. This training included assessment activities for the participants to receive feedback on their level of understanding of the framework. Next, the participants were trained on how to use ALApp, including registering their classes in the app and recording data. In the first cohort of students (a total of four cohorts have been through the two studies), training was done using an in-person format. This proved tedious, however, given the varied and busy schedules of the students. Subsequent cohorts were provided all training through on-line modules embedded in the university's learning management system.

In the first study orientation meeting, students were given instructions on how to install and register in the ALApp. Fig. 1(a) shows the user interface that a student used to select the classes in which they are enrolled. Based on this information, ALApp would send a push notification to each student at the end of every class period. The student was then expected to enter the ICAP data as shown in Fig. 1(b) for the Android user interface. Similar screens were created for iOS.



(a) Class Selection. (b) ICAP Data Entry.

Fig. 1: Actively Learning Android App Screenshots.

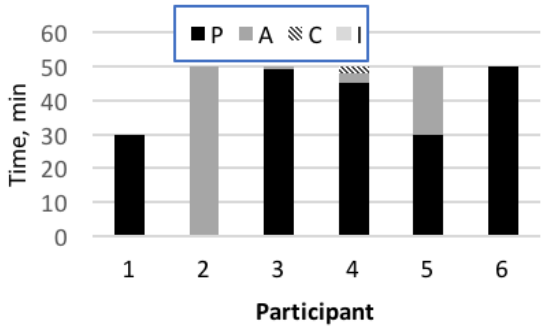
Students were expected to enter ICAP data for all activities, whether in or out of scheduled class times. Out-of-class activities might include things like completing homework, visiting a professor's office hours, or attending a group project meeting or study session. In order to help students remember to enter these non-class activities, the ALApp sends notifications every three hours between 10 am to 10 pm (inclusive), unless the student was already scheduled for a class at that time.

Validating that students were consistently entering their ICAP data was initially done manually by someone who was paid to look at the ALApp database every week. If they found students who had missed two consecutive class activity entries, they would report them to the Principal Investigator (PI) who would send reminder emails to the student. This was very tedious work. Scripts were then written to automate this checking and emailing, with copies going to the PI.

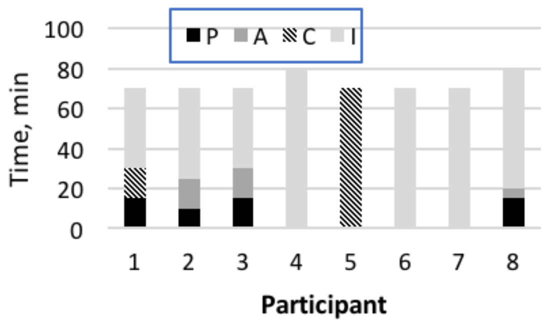
III. Early Study Results

To demonstrate the use and accuracy of the ALApp, Fig. 2 shows data taken from students in 50-minute class meetings in two different introductory thermodynamics courses [5]. In Fig. 2(a) data from six participants show that the instructor in this class employs a largely lecture style of teaching and, in turn, the students uniformly report active and passive modes of activities. In the other class (Fig. 2(b)), the instructional mode is consistently dominated by constructive and interactive activities among the eight participants.

In each mode of instruction shown in Fig. 2, there is still substantial student-to-student variation among each student's perceived level of engagement. The question is whether these are true variations or poor training on the ICAP framework,

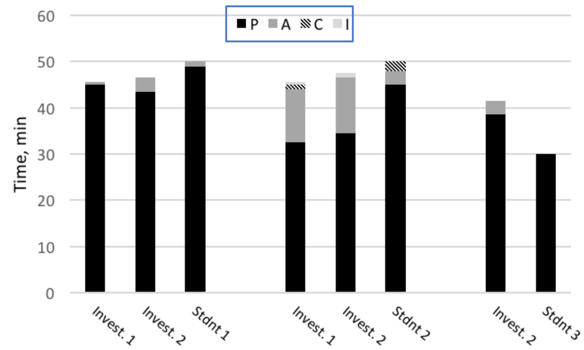


(a)

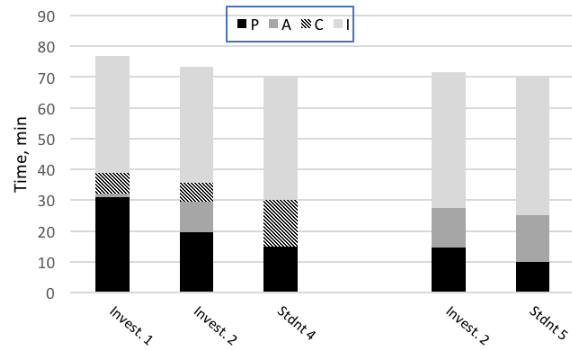


(b)

Fig. 2: ICAP data from participants in a (a) lecture-based and (b) activity-based class.



(a)



(b)

Fig. 3: ICAP data from observers of participants in a (a) lecture-based and (b) activity-based class.

which leads students to assign times to incorrect categories. To test this, and to validate the accuracy of the data entries, two investigators attended randomly selected classes and measured the ICAP times of some study participants through direct observation (recall that the ICAP framework relies on overt, observable behaviors). The participants were told beforehand that the investigators were attending class to observe “a typical class,” but not that any of them were being directly observed. Fig. 3 shows comparisons between the investigators’ and each student’s measurements from each class.

In general, the agreements between the two investigators and between the investigators and each student are very good. All measurements correctly capture the predominant instruction mode of each class. Fig. 3 further demonstrates that, indeed, there are variations between students in their level of engagement during the same class. Furthermore, these variations are largely confirmed by the investigators’ observations.

IV. Software Architecture

A primary contribution of this study is the software application that was developed for collecting student data. This section describes ALApp’s software architecture which involves the selection of software technologies and their organization. Software technologies evolve rapidly, with new technologies emerging frequently. As a result, selecting a robust, secure software architecture that will remain stable over time and can evolve to support unforeseen features is both challenging and essential.

1) Technology Stack: Fig. 4 shows the organization of the primary software technologies used to create ALApp. The open source Parse Platform powered the ALApp. Parse Platform provides native iOS and Android SDKs and a push notification service. The Parse Platform was chosen over alternative services for having native SDKs, a hosted cloud service, and a generous free tier. The iOS app uses the standard iOS SDK and Swift. Android uses the standard Android SDK and Java. Developing native iOS and Android apps was chosen over using a cross-platform tool, such as Xamarin, based on the development team’s existing expertise. Additionally, push notifications to non-native mobile web solutions had significant restrictions compared to native apps. After the hosted Parse Platform was shut down by Facebook, we hosted our own Parse Platform instance on a Linode server.

2) Database Schema: Fig. 5 shows the database schema implemented in Parse for the ALApp. Parse Platform uses MongoDB as its backend database. Arrows represent a Parse Platform reference from one table to another, called a Pointer. The Installation table keeps track of the UUIDs required to send push notifications. The RegCodes table contains the list of approved codes to log students into the ALApp. The AvailableClasses table contains the list of classes from which

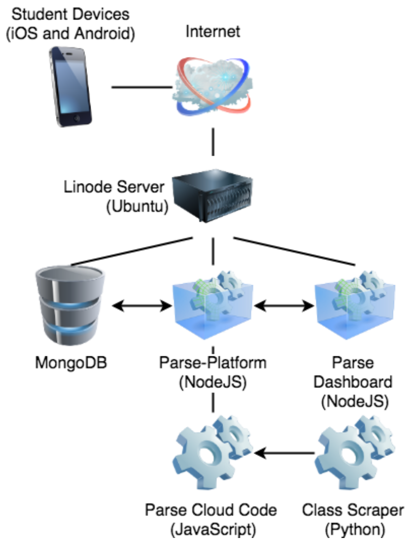


Fig. 4: Actively Learning Software Architecture.

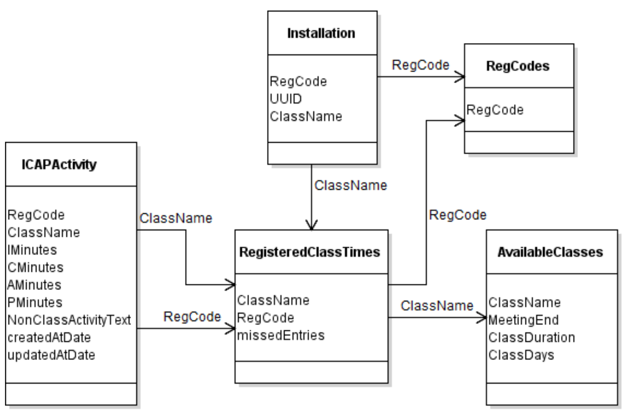


Fig. 5: Actively Learning Database Schema.

students choose. RegisteredClassTimes holds the list of classes students selected, and the association with a particular student. ICAPActivity holds the recorded student activity data.

3) Cloud Code: ALApp uses Parse Cloud Code's Parse JavaScript SDK to interact with the database on the server side. Cloud Code powers the push notification service, email reminder system, and a Python-based web scraper to automatically populate class data for the AvailableClasses table. Cloud Code functions ensure push notifications are sent accurately and according to the specified schedule. Functions also monitor the database to send the automatic email reminders to students, and ingest the class data from the Python scraper.

V. Personnel Considerations

Education research often occurs in the context of university faculty using grant funds to hire students to complete the work. Because a major component of this project involved software development, a computer science faculty was included as a co-PI who was tasked with hiring and directing undergraduate students to develop the ALApp. Hiring computing students has proven to be very challenging due to the strong job/internship market outside the university, and institutional policies that limit hourly wages inside the university. The PIs hired three computing students to implement different aspects of ALapp as their senior project, which is required of every student. A fourth second-year student was also hired before he had the opportunity to get an internship. This approach carried the risk of students completing a small portion of the project during their six-month senior project, then leaving the project for someone else to learn, extend, and maintain. Mobile and web development technologies are continually evolving, requiring on-going maintenance activities even if no features are being added to the applications. Fortunately, one of the senior project students agreed to stay with the project after graduation, working on it part-time to the present. The combination of this former student and a computer science co-PI has provided the consistency needed to keep the system relevant.

VI. Institutional Challenges

One of the key features of ALApp is the ability to know a student's schedule so they can be notified when they get out of class to remind them to enter their ICAP data while it is fresh in their mind. The development team originally hoped to pre-populate every student's course schedule by interfacing with the university enrollment information system. Although this was technically possible, the University Privacy Officer determined that this involved too much risk. The development team pivoted and implemented scripts that scraped course schedule information from a publicly available website, and exposed this data to students who could then manually select their course schedule as shown in Fig. 1(a). This proved to be an excellent solution that kept student identities anonymous in ALApp with only minimal burden on the student to select their courses at the beginning of each academic term.

VII. Lessons Learned

At the onset of the project, Parse was a very popular platform for quickly developing and deploying mobile apps. Parse was recently acquired by Facebook which increased Parse's credibility and likelihood to stick around. To the surprise of many, Facebook announced in January 2016 that it was closing down Parse by January 2017. ALApp was already implemented and tightly integrated with Parse, so there was no time or budget to rework ALApp to use a different backend technology. Fortunately, Parse released an open source version which others released as Parse Server. This may have been a blessing in disguise because it allowed the team to install Parse Server on it's own Linode server and minimized the required technology changes to keep ALApp current.

A second lesson learned was that the Android and iOS app stores are not designed for research apps. The team did not want to make the apps available to the public, but still wanted to make them available to student participants through the app stores. TestFlight for iOS and the Play Store beta testing feature for Android were used to enable relatively simple install access for the students, although TestFlight requires manual effort to renew and deploy quarterly.

References

[1] Duckworth, A.L., Peterson, C., Matthews, M.D., and Kelly, D.R., "Grit: Perseverance and passion for long-term goals," J. of Personality and Social Psychology, 92(6), 2007, pp.1087–1101.

[2] Duckworth, A.L., Quinn, P.D., and Seligman, M.E.P., "Positive predictors of teacher effectiveness," The J. of Positive Psychology, 4(6), 2009, pp.540–547.

[3] Duckworth, A.L. and Quinn, P.D., "Development and validation of the short grit scale (Grit-S)," J. of Personality Assessment, 91(2), 2009, pp.166–174.

[4] Duckworth, A.L., Kirby, T.A., Tsukayama, E., Berstein, H., and Ericsson, K.A., "Deliberate practice spells success: Why grittier competitors triumph at the National Spelling Bee," Social Psychological and Personality Science, 2(2), 2011, pp.174–181.

[5] Chen, J., Janzen, D., McGaughey, K., & Widmann, J., "Tracking learning engagement at the student level," Proceedings of the 2019 Frontiers in Education Conference, Cincinnati, OH, USA, 2019.

[6] Chi, M.T.H., "Active-Constructive-Interactive: A conceptual framework for differentiating learning activities," Topics in Cognitive Science, 1, 2009, 73-105.

[7] Fonseca, B.A. & Chi, M.T.H., "Instruction based on self-explanation," Ch. 15 in Handbook of Research on Learning and Instruction, R.E. Mayer and P.A. Alexander (Eds.), 2011, New York, NY: Routledge.

[8] Prince, M., "Does active learning work? A review of the research," Journal of Engineering Education, 93(3), 2004, 223-231.

[9] Felder, R.M. & Brent, R., "Active Learning: An Introduction." ASQ Higher Education Brief, 2(4), 2009.

[10] Prince, M. & Felder, R.M., "Inductive teaching and learning methods: definitions, comparisons, and research bases," Journal of Engineering Education, 95(2), 2006, 123-138.

[11] Prince, M. & Felder, R.M., "The many faces of inductive teaching and learning," Journal of College Science Teaching, 36(5), 2007, 14-20.

[12] Chen, J., Janzen, D., Widmann, J., McGaughey, K., & Teramoto Pedrotti, J., "Measuring active learning at the student level," Proceedings of the 2017 Research in Engineering Education Symposium, Bogota, Colombia, 2017.