Collaborative Design and Build Activity in a CS1 Course: A Practical Experience Report

Géraldine Brieven[1], Laurent Leduc[2], Benoit Donnet[1]
[1]Institut Montefiore, Université de Liège, Belgium, [2]IFRES, Université de Liège, Belgium.

Abstract

Shaping students' mind to structure and solve problem, in an Introduction to Programming course for first year students, takes time, leading some students to get demotivated before they actually master this new skill. This paper reports on a Collaborative Design and Build activity dedicated to reinforce students' interest and improve their skills by sequentially solving problems in teams, through a real-life inspired scenario. Two sessions of the activity were organized during the semester. This paper carefully describes the activity design. It also presents and discusses results showing that students' productions got more accurate across the sessions, leading many participants to outperform in solving problems in the final exam.

Keywords: *Collaborative Learning ; Role Taking ; First Year Students ; Computational Thinking ; Active Pedagogy ; Clustering.*

1. Introduction

A main objective of our course "Introduction to Programming" (usually referred to as "CS1") consists in teaching to first year Computer Science students how to design a solution before actually building it. Many students find that reflection phase (called "Design Phase") hard, abstract, and tedious. In practice, as soon as they are working on their own, students tend to rush or even skip the design to directly jump to the implementation (perceived as more funny). That leads to a poor solution they patch afterwards, resulting in a final shaky solution. To tackle it, we need to assist more closely the students during the Design Phase, to emphase its importance (Harackiewicz, 2016) and "unlock" their understanding. This paper describes a Collaborative activity, incorporated in our CS1 course, relying on Role Taking (included in Collaborative Learning) (Strijbos & Weinberger, 2010). The activity highlights the purpose of the Design Phase and gives students co-ownership. It is hold for groups of six students whose goal is solving three problems, by first designing their solution, in a limited amount of time. For a given problem, the resolution draws on an assembly-line process, following the concept of Assembly Line Education ("ALE") experimented in the Medicine Faculty in University of Florida where it has been demonstrated that this teaching mode increases interest and sparks students (Rosario et al., 2020). By acting so, the students need to give their best since they are laying the foundations for other teams. Moreover, they get instantaneous feedback from each others, making them aware of their deficiencies in their ability to design a solution. This *Collaborative Design and Build activity* (abbreviated here as "CDB") was also set out to be as close as possible to a real professional scenario in order to bring more sense to the solution design process and stimulate students (Bédard & Béchard, 2009). Eventually, it is aimed to see students producing higher quality solutions, relying on the vision that interest energizes learning (Hidi, 2000).

2. Context of the CS1 course

From a content perspective, the course alternates between specific C programming concepts and a problem-solving methodology, inspired by the idea "Thinking before Implementing" (Razzouk & Shute, 2012). It consists in learning the students to handle complex problems by first designing their solution (through the *Design Phase)* so that, eventually, implementing it (through the *Building Phase*) gets straightforward. The Design Phase relies on Computational Thinking concepts (Rodríguez del Rey et al., 2020) and includes two sequential steps:

1. *Problem Analysis*. It consists in defining the problem through input and output and, then, decomposing it into several subproblems easier to handle than the original one.
2. *Graphical Reasoning* It consists in graphically representing the iterative process associated to a given subproblem so that this graphic will support the code (Liénardy et al., 2020).

Then, the Building Phase comes, made up of the last step :

3. *Coding*. It consists in writing the code (in C programming language) based on the two previous steps and testing it with respect to the initial problem.

In practice, during traditional sessions where students resolve exercises, in isolation to each others, they often find hard, tedious and useless to go through this process for easy problems. But as soon as problems get more complex, they are not able to utilize properly the methodology and their final solution gets very poor due to a too superficial Design Phase. This phenomenon has been periodically observed in recent years, in particular during the MidTerm evaluation (that is the first certificative assessment the students have to take).

To tackle this recurrent trend, the *Collaborative Design and Build Activity* (i.e., CDB) was created and proposed twice to the students, in order to emphase the importance of the Design Phase and make them involved and rigorous across this process. Fig. 1 depicts when the two CDB sessions were organized with respect to other usual class activities and evaluations. More precisely, this figure shows that the course is made up of theoretical and practical sessions (where only the ones related to the problem-solving methodology are highlighted). Moreover, a MidTerm evaluation is organised, following similar conditions than the final Exam, with a strong emphasis on our methodology (both Design and Building Phases).

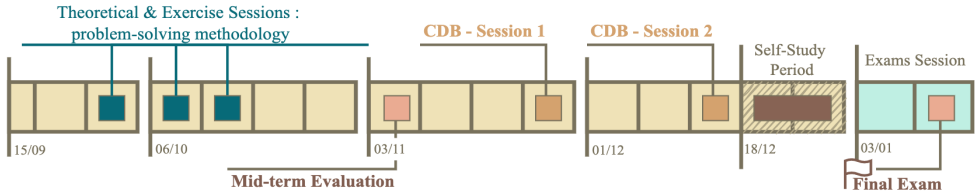


Figure 1. Inclusion of the CDB activity in the CS1 course organization.

3. Organisation of the CDB Activity

The CDB activity has been set up for a mid-size group (around 50 participants) and could afford until 100 participants with two supervisors. Here, 50 students participated to the first CDB session while only 25 took part in the second one. 23 students took part in both sessions.

The right side of Fig. 2 ("Classroom Configuration") shows that the CDB activity is experimented by *Groups* of six students, each Group being split in three *Teams* (two students per Team). The goal of each Group is to solve three problems in a limited amount of time. The left part of Fig. 2 illustrates how the three problems are getting progressively solved in parallel over time, following the three steps of the methodology taught in the course.

This conception is inspired by real professional life as, in large-scale development projects, the three steps of the methodology are performed by different computer scientist teams.

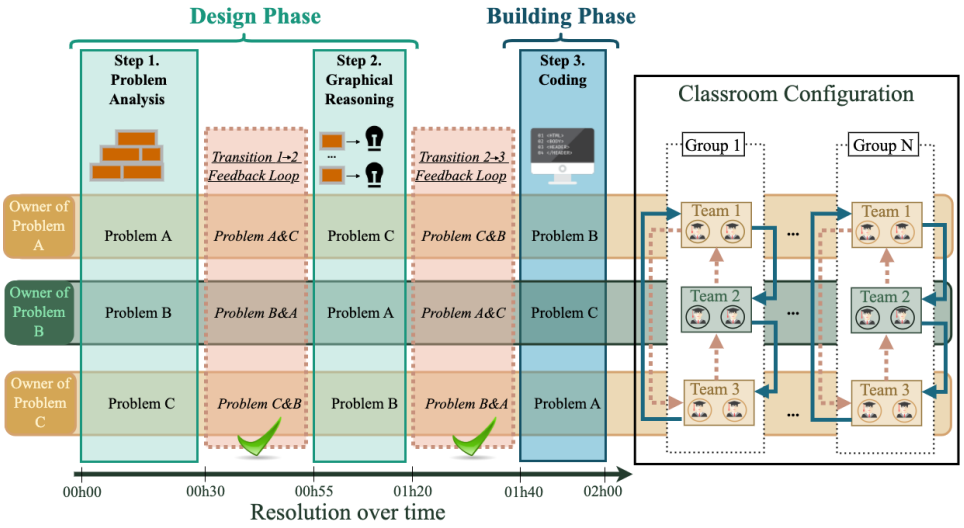


Figure 2. Teaching Scenario Organization.

Going into more detail, at the beginning, each Team receives and gets responsible of one problem. For a given problem, the three steps are sequentially addressed, each Team being busy, turn-by-turn, with a specific one. At the end of each step, each Team work moves to the next Team, clockwise, in the same way in every Group, as shown by the plain arrows.

For every step transition, a validation period is dedicated to allow the students to read the production provided by the previous Team and, if needed, report a feedback to them, as depicted by the dashed arrows. The motivation behind this is to limit the impact of a "poor quality work" on the next productions that are based on the previous ones. This feedback is supported by a validation form listing the criteria the production should meet to be a good ground for the next step. Once filled, this validation form can be returned to the previous Team who should adapt their work based on the comments in the form.

Once the Coding step is over, every Team checks if the final code related to the problem they were the owner of correctly answers the initial question. Then, the three Teams join together and share their conclusion. Finally, they pick one solution that was not working (if there is any) and identify the various error sources. This last investigation work is supported by a dedicated template listing the main requirements the solution should have met. This template is kept flexible, allowing students to easily add specific observations.

4. Method

Data was collected to evaluate the impact of the CDB activity on the students' perception and performance. In particular, it is aimed to see how much students' learning got supported through Collaboration, with a dedicated focus on their performance on the Design Phase.

4.1. Perception Data

An anonymous survey was administered to students at the end of each CDB session. Both surveys included open-ended and Likert scale questions, asking how they felt during the activity, what they learnt, which difficulties they encountered, and how communication went.

4.2. Performance Data

For each CDB session, all students' productions were collected and labeled by their University identifier in order to be analysed afterwards. In this way, the list of participants to each session could be derived. We consider in our performance analysis the students who took part in at least one session. Moreover, the level of each pair of participants (i.e., each Team) is qualified by rating their production through a score value that is mapped to a cluster.

Figure 3. Mapping from their score to a cluster.

As shown in Fig. 3, the absents get a score of -1 for discriminating poor performance and not taking part in the CDB activity. The remaining clusters are assigned to students based on score intervals that are evenly space sample from 0 to 5. A score of 0 means that the Team has performed a poor job, while a score of 5 means the production is correct. To compute the score, some typical mistakes are defined and each of them is mapped to a gravity factor. The largest the factor, the most serious the mistake and the more penalized the students if they fall into it. Those errors are categorized as syntax error or semantic error. This classification is relevant when the second and the third steps are assessed since they rely on some primary work from other Teams. That means that the correctness of those last steps might be badly impacted by the previous Teams, independently from any lack of knowledge and skills from the current Team. Under that concern, if the input coming from the previous Team is qualified as too poor, only the syntax errors are assessed to derive the score.

5. Results and Discussion

5.1. Students' Perception

To the question *"Did you like the activity ? Why ?",* roughly 88% of the participants in the first session answered positively, with 39% of them spontaneously plebisciting the fact that this was a group activity and 16% this was a real life inspired activity. Fig. 4. also shows learning outcomes of the CDB activity, according to participants. A large panel of aspects came up, especially at the end of the first session since it was completely new to them. However, after the second one, many students felt they learnt more about the methodology

itself (mainly the 2 first steps)**.** This last observation shows the importance of repeating the CDB activity throughout the semester in order to really bring improvements in their ability to handle the Design Phase. This inference was enforced by the participants themselves. At the end of the second session, from the statement *"The quality of the work you got from other teams improved compared to the first session.* ", 63% of them agreed or even strongly agreed.

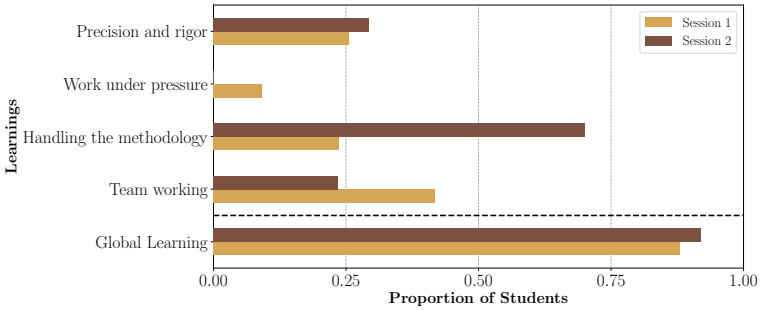


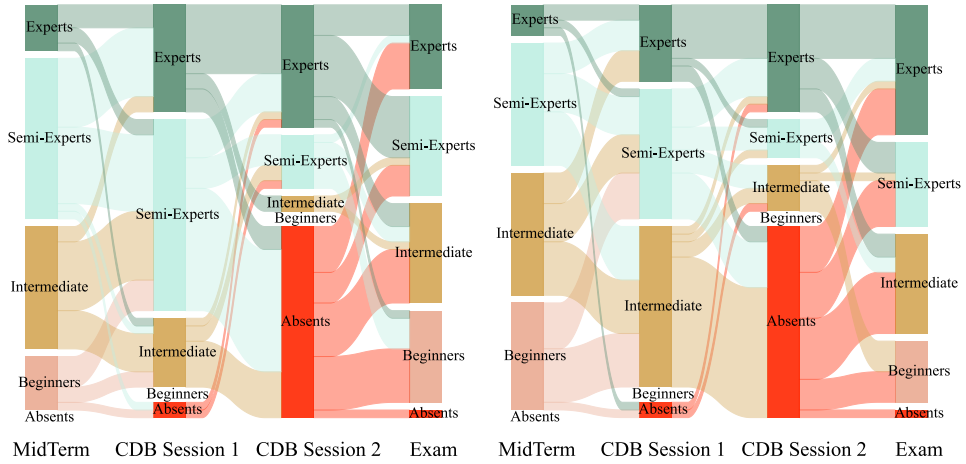
Figure 4. How much and what students have learnt in particular during the CDB activity (Answers to : "Have you learnt something in the CDB session? If so, what in particular?").

5.2. Students' Performance

We focus here on the performance over the Design Phase, targeted to be promoted by the CDB. Then, more broadly, the resulting performance on the Building Phase is given as well.

In order to qualify the impact of the CDB activity on students' performance, two end references framing the two CDB sessions are defined : the MidTerm and the Exam. For all those four activities, the students were assessed based on the same criteria. However, some intrinsic differences remained, leading to overestimate the students' performance during the CDB sessions compared to the evaluations. In particular, during the CDB sessions, there was (*i*) less stress, the CDB activity having been introduced as a formative exercise; (*ii*) the opportunity to roll backward thanks to the feedback reported by other Teams; (*iii*) more time to handle each step; (*iv*) collaboration, which can be considered as both reinforcement and additional difficulty to handle, depending on the soft-skills and the knowledge of each student; (*v*) a focus on the methodology, while the evaluations also cover (many) other topics.

Fig. 5 illustrates the distribution and evolution of the participants over the various clusters. In particular, Fig. 5a provides the mean results over the whole Design Phase. It shows that many participants improved from the MidTerm to Session 1. More precisely, we computed that 52% of them moved from one cluster to an upper one. Then, 20% of the participants got even better in Session 2, half of them growing from Semi-Experts to Experts. However, a large rate of Absents is also observed for Session 2. A likely reason is that the CDB sessions ran (too) late over the semester (as shown in Fig. 1), since it got organized at the same time as the idea rolled forward. In Session 2, all the time budget dedicated to the course activities



5a. For the Design Phase (Step 1+ Step2). *5b. For Graphical Reasoning (Step 2).*

Figure 5. Distribution and evolution of the participants across clusters over the activities.

was consumed. Moreover, the session was not promoted enough, leading many students to prefer starting their studying period earlier, rather than taking part in another CDB session.

With respect to the Exam, one can notice that both the Experts and Beginners clusters grew compared to the MidTerm. To catch what is behind the larger proportion of Beginners, we have considered separately the 2 steps of the Design Phase. Taking that detailed perspective, we could link that performance decrease to the Analysis step that was addressed slightly differently in the exam, which disrupted some students. However, the Graphical Reasoning Step (roughly independent from the Analysis one here) got successfully handled by many students. More precisely, from Fig. 5b (complemented by the corresponding computated percentage), we can see that the proportion of Semi-Experts joined to the Experts rose (by 40%) from the MidTerm to the Exam while the proportion of Beginners dropped (by 43%). That improvement was likely boosted by the CDB, in accordance with the observation that Collaborative learning can lead to better development of ideas and concepts through sketch drawing (helping students in sharing the same representation) (Komis et al., 2002).

Finally, shifting to the Building Phase, we notice that the group of Experts and Semi-Experts grew from 3 to 23 participants from the MidTerm to the Exam. Again, it enforces the idea the CDB activity is a good springboard to make students embracing the Design Phase.

6. Perspective and Conclusion

From this first experience, CDB appeared to be a cross-skilling exercise, learning students to handle intra and inter-team communication, be precise and rigorous, and improve their problem-solving skill. During both sessions, students were stimulated by working on a real-

life scenario where they needed to give their best to avoid penalizing the others in performing their own task. This dependence also gave the opportunity to put in place feedback loops, learning students to provide feedbacks and adapt from them. Besides those positive aspects, we believe the CDB activity could be better tuned. In particular, it should be better articulated with the rest of the course, e.g., the first session should come prior to the MidTerm and any session should be better advised to ensure students' participation. Moreover, the first CDB session could be made mandatory, so as to make students realizing early enough the interest of the Design Phase.

More generally, we believe Collaborative Learning and Assembly Line Education (both encapsulated in a real-life scenario) have the potential to make the difference in students' engagement and performance.

References

Bédard, D., Béchard, J.-P. (2009). Innover dans l'Enseignement Supérieur. *Presses Universitaires de France.*

Harackiewicz, J. M., Smith, J. L., & Priniski, S. J. (2016). Interest Matters: The Importance of Promoting Interest in Education. *Policy Insights from the Behavioral and Brain Sciences*, 3(2), 220–227. doi: 10.1177/2372732216655542

Hidi, S. , Harackiewicz, J. M. (2000). Motivating the Academically Unmotivated: A Critical Issue for the 21st Century. *Review of Educational Research Summer*, (70)2, 151-179. doi: 10.3102/00346543070002151

Komis, V., Avouris, N., Fidas, C. (2002). Computer-supported collaborative concept mapping: Study of synchronous peer interaction. *Journal of Education and Information Technologies*, 7(2), 169-188. *doi: 10.1023/A:1020309927987*

Liénardy, S., Leduc, L., Verpoorten, D., Donnet, B. (2020). CAFÉ: Automatic Correction and Feedback of Programming Challenges for a CS1 Course. *22nd Australasian Computing Education Conference (ACE),* 95-104. doi : 10.1145/3373165.3373176

Razzouk, R., Shute, V. (2012). Why is Design Thinking and Why Is It Important ? *Review of Educational Research*, 82(3), 330-348. doi: 10.3102/0034654312457429

Rodríguez del Rey, Y. A., Nissandra Cawanga Cambinda, I., Deco, C., Bender, C., Avello-Martinez, R., Villalba-Condori, K. O. (2020). Developing Computational Thinking with a Module of Solved Problems. *Computer Applications in Engineering Education*, 29(3), 506-516. doi: 10.1002/cae.22214

Rosario, J., Lebowitz, D., Webb, A. L., Ganti, L., Vera, A., Macintosh, T., Walker, A., Rubeor, J. (2020). Assembly Line Education: A Novel Educational Technique for Today's Learners. Cureus, 12(2). doi: 10.7759/cureus.7065

Strijbos, J.W., & Weinberger, A. (2010). Emerging and scripted roles in computer supported collaborative learning. *Computers Human Behavior*, 26(4), 491-494. doi : 10.1016/j.chb.2009.08.006