World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:15, No:3, 2021

Co-Creational Model for Blended Learning in a Flipped Classroom Environment Focusing on the Combination of Coding and Drone-Building

A. Schuchter, M. Promegger

Open Science Index, Computer and Information Engineering Vol:15, No:3, 2021 publications.waset.org/10011931.pdf

Abstract**—**The outbreak of the COVID-19 pandemic has shown us that online education is so much more than just a cool feature for teachers – it is an essential part of modern teaching. In online math teaching, it is common to use tools to share screens, compute and calculate mathematical examples, while the students can watch the process. On the other hand, flipped classroom models are on the rise, with their focus on how students can gather knowledge by watching videos and on the teacher's use of technological tools for information transfer. This paper proposes a co-educational teaching approach for coding and engineering subjects with the help of drone-building to spark interest in technology and create a platform for knowledge transfer. The project combines aspects from mathematics (matrices, vectors, shaders, trigonometry), physics (force, pressure and rotation) and coding (computational thinking, block-based programming, JavaScript and Python) and makes use of collaborative-shared 3D Modeling with clara.io, where students create mathematics know-how. The instructor follows a problem-based learning approach and encourages their students to find solutions in their own time and in their own way, which will help them develop new skills intuitively and boost logically structured thinking. The collaborative aspect of working in groups will help the students develop communication skills as well as structural and computational thinking. Students are not just listeners as in traditional classroom settings, but play an active part in creating content together by compiling a Handbook of Knowledge (called "open book") with examples and solutions. Before students start calculating, they have to write down all their ideas and working steps in full sentences so other students can easily follow their train of thought. Therefore, students will learn to formulate goals, solve problems, and create a ready-to use product with the help of "reverse engineering", cross-referencing and creative thinking. The work on drones gives the students the opportunity to create a real-life application with a practical purpose, while going through all stages of product development.

Keywords**—**Flipped classroom, co-creational education, coding, making, drones, co-education, ARCS-model, problem-based learning.

I. Introduction

THERE is an increasing demand for people working in the technical field and especially in engineering and IT. To integrate scientific and engineering knowledge into society, the STEM initiative was founded in Germany. STEM (German: MINT) initiatives focuses not only on teaching computational thinking at schools and universities but also to people without any educational background in IT. Their aim is to increase the rate of women and to counteract the skilled labor shortage in mathematical, technical, scientific and engineering fields. Despite all of these efforts, we tend to neglect the haptic aspect of IT and forget the art and fascination of understanding something by using our hands. This also reflects in our use of language: an English term for "to understand" is "to grasp", in German we use a similar term - "begreifen" - that includes "greifen" (to grasp). Touching something is an integral part of our thought process when we try to understand new concepts. The use of our hands and the feeling of building something real and tangible should be a part of our educational methods [1]. With coding, this combination is possible. This paper proposes a different, more hands-on approach to this topic, using programmable drones as a showcase for programming. When diving deeper into real-life applications, it is crucial for children and students to understand the relevance and potential uses of what they are learning. Compared with learning in a theoretical way, practical learning, earns better results. It is a promising approach with a wide range of applications that should be included in more primary school curricula [1]-[8]. What makes this approach unique is that, firstly, we use the same model in a playful way with block-based programming as well as with pure script language. Our concept gives students the possibility to interact with their results in a practical way, so programming is not just a theoretical software design; it also has an impact on the functionality of hardware. Another vital aspect of our teaching model is that we guarantee that students feel a sense of achievement, which in turn, increases their interest and motivation for IT [1], [7], [8].

The entire project also aims to spark girls' interest for IT topics and creative building processes. Moreover, we would also like to use our model to help recruit prospective university students. With our approach we have shown that students of all ages, backgrounds and genders can be fascinated by coding and their learning outcome is greater when we combine theoretical and practical approaches [1]. Even when doing an online session, this combination works. Online education has become a vital part in our lives. It challenges us to create a knowledge transfer, to support the students and to interact with them. Traditional teaching methods are getting more and more neglected, and flipped classroom environments, where students are placed into an active learning environment, are getting more important nowadays. In order to accomplish the best teaching effect, we

Arthur Schuchter is with the University of Tromsø, UiT i Bodø, Nord Universitet, PB 143,Universitetsalléen 11,8049 BODØ, Norway (phone: 0043 660 5718299; e-mail: asc034@ uit.no).

Marianne Promegger is with the Gasthofsiedlung 100, A-5531 Eben im Pongau (phone: 0043 660 4580500; e-mail: marianne.promegger@ph-ooe.at).

have to find ways to collaborate in teams or pairs. Many companies need employees with co-working skills as well as people who think outside the box, i.e., who see different ways for problem solving. Our educational system is strongly focused on non-collaboration and we tend to work together only on some occasions. When finally entering university, there is a need for working together since students have to solve bigger problems in an embedded system. When ultimately entering the career world, we have to collaborate, we have to start reverse thinking and we have to solve problems; therefore, it is vital to be prepared [7], [9], [10]. However, technical fields are increasingly software-based and even traditional fields such as car or plane manufacturing are now dominated by computers and software, instead of operators using screws. Traditional handicraft and manual labor are being neglected and deteriorate slowly. Branches such as cobblers or hat makers are already rare professions and their number continues to decrease. We have forgotten how to do the most basic tasks such as screwing a shelf to the wall or sewing on a button, either because we have never learnt it how to do it or because we never had to. In our world of commodity, we rather pay someone to do these things for us instead of having to do everything ourselves. We believe that there is simply no need to have all of these skills. Children have trouble learning how to write with a pen because they are used to playing with screens on smartphones and tablets. Our fine motor skills are crippled by our daily reality of pushing buttons and using voice commands. We forget that all software we program cannot function on its own – it needs hardware to run on, and this hardware has to be built and maintained [4]-[6], [12], [14], [19].

In order to adequately prepare students for future careers in IT, we have to take a look at how programming is being taught in our schools. Coding and programming are currently included in the Norwegian elementary school curriculum through Lego-league projects and apps for coding as well as "mechanical" coding, where students code together in the classroom. However, school courses on programming usually already attract kids interested in the subject.

Blended learning in a flipped online classroom is a method of teaching, where digital media and technology are combined with regular teacher-led classroom assignments. With this combination, students have more versatility to assimilate their experience of learning. The students are acquiring the contents independently via online videos with the model of blended learning in a flipped classroom. The class time is then used for "homework", or the classroom itself is redesigned in a way that the teacher is no longer the focal point. Teachers no longer spend class time delivering direct instructions but use it to guide supervised practice and provide individual assistance where needed [9], [10].

We go an ambitious way in re-designing traditional flipped classroom models by putting the student more into a role of a content creator and educator. In traditional flipped classroom models, obtaining of new content mostly occurs through watching videos and transferring information from instructor to students utilizing technological tools. We have developed this idea further and we put the students in a more active role, where they create something in teams and in projects. Our approach is even more ambitious as we strive to combine knowledge transfer for coding, math, physics and 3D modeling, focusing on building a flyable drone. The whole module lasts 3-4 months and is spilt up into different parts. We have created several ways of achieving co-creational content generation in flipped classroom problem and project based learning environments [12], [14]-[18].

With the ARCS model, which means attention, relevance, confidence and satisfaction, we find that motivation is an important factor in class. Other key aspects are reverse engineering and flipping the process [6], [9], [10]. First, the finished product - in our case the drone - will be shown, and afterwards it will be dissembled step by step. In order to succeed with reassembling it again afterwards, it is important to write down all the steps. With the help of the instructor, principles will be explained and illustrated [6].

Once students grasp how it works, they can help other students or lead small groups and are therefore acting as a multiplier. There is a wide range of activity in the higher education sector labeled "students as partners" and "co-creation in learning and teaching". Several frameworks have been proposed to map and categorize existing partnership and co-creation roles, activities, research, and practices [18]. In this paper, we synthesize some of these frameworks to illustrate how the predominant focus is already utilized (in international literature) in some of the partnership projects with only a few chosen students, who are already preoccupied or favored. On the contrary, the co-creation of teaching and learning, within a regular class setting involving the curriculum, has passed to oblivion [16]-[18].

We explore the potential of co-creating, learning and teaching with a whole class of students (including face-to-face, blended and online settings, as well as lectures, tutorials, laboratories, and other methods of teaching); in other words, it is a co-creation which is integral to students' programs and courses of study. We argue that whole-class approaches to co-creation may be inherently more inclusive of students than others. With this idea the relationship between teachers and students and classmates with classmates is greatly enhanced.

Collaborative learning has become a big part of our lives. Still, cooperative work between students in different learning atmospheres (for example in the classroom) is way more compound. In order to fully grasp the idea of all the various points of views, we must understand the social, motivational, cognitive, contextual, and emotional aspects of collaboration. Working together in a dynamic and systemic team setting is vital. With our approach we have shown that students of all ages, backgrounds and genders can be fascinated by coding and that their learning outcome is greater when we combine theoretical and practical approaches [12], [14]-[18].

II. State of the Art of Co-creational Learning

Co-creation means establishing a deeper connection between a teacher and a student, and also between a student and other students. The main point is to apprehend education as a shared attempt where teaching and learning is accomplished with the students and not to the students. Co-creation intersects with active learning, which means focusing on applying an active role, versus a passive role, for the collaboration with student and teacher [12], [14]-[18].

Although active learning definitions and practices may differ, they all involve the students' interaction, contribution (physical and mental) and participation in activities to get impartations, reflection on one's own knowledge and problem solving. Attitude and merits investigations play an important role in activities that involve discussions, writing or reading, or small group work. Elements such as the purpose of their work, or the negotiation of the content of the subject, or even the teaching approach, can enhance their working and learning process immensely. Students can figure out their favorite approach for an assessment and all the different ways they can learn and work in a team [16]-[18].

Every successful organization runs smoothly with collaboration in the workplace. Working in a team has different facets of organization. One of the most important qualifications is being able to work in a collaborative environment. If not already practiced in school, it will become even more important in the workplace. A precise given structure is vital in order to succeed in a project. Communication is key among the team players. Especially in science, engineering and upmost in IT, collaborative working is inevitable. Companies nowadays are looking for people who can give new impulses, think innovatively and especially think in reverse. In order to understand how something works, you have to implement reverse thinking, which means going back step by step, starting from the end (end product) and working your way back to the beginning [12], [14]-[18].

There are five simple steps in order to succeed with creating a culture of collaboration in the workplace. First, you need to have a clear vision. Second, you have to recruit and establish collaborative leaders that support the company's vision and can lead a team. Third, it is important to give opportunities for collaboration. To learn from the other team players, which ideas and capabilities they have, is necessary in order to grow. Fourth, you have to use social collaboration tools, such as cloud-based tools and internal social networks, that are a great way to communicate, especially when working in teams with varying geographic locations and different time zones. The last point is to steadily reinforce the goals in order to succeed [12], [14], [15].

If students are already familiar with such a strategy because they learnt it at school, they will be well prepared for the working world. When implementing co-creation in a company, it is vital to bookmark collaborative culture. It requires creative thinking in order to solve problems, institutionalized and experiential learning, leadership, quality management, and communication with constant improvement to grow steadily. It is a learning journey [12], [14], [17], [18]. It is critical for the social aspects, since trust and mutuality are involved, and common understanding is important. Especially online sessions need special attention to a sophisticated set of variables, which involves emotional, contextual, cognitive, social, and motivational challenges [12].

For students, co-education and co-working, which are a new form of work, matter. Online sessions gained even more on significance during the pandemic. Online teaching is not only a nice add-on feature; it is an essential modern teaching tool. The knowledge transfer as well as the knowledge acquisition is absolutely necessary. Especially in physics and math, online teaching with flipped classroom and problem-based models are already widely spread and used [9], [10], [15]-[18].

The book "Co-creation in Higher Education" mentions how we have to prepare our children for the future, even though we do not know how it will look like. It is paramount what will happen in the education system, in the private as well as in the public one. Whereas usability is being taught in lectures, life is happening intermediately. The relevance of relationships, the relevance to the questions of our time and to the society in which we live in, as well as the relevance of knowledge to our lives is key [15], [18].

III. The Coding Club Initiative

Typically, Coder Dojos or Code Clubs are free IT communities helping others learn to code and see what fun things can be done in information technology. Such clubs have been a popular part of the IT world where people mingle to meet with other data nerds, exchange their ideas and build a community. Our approach to Code Clubs is slightly different:

The Coding Club (*codingclub.at*) has now been active in Austria for more than three years. The idea was originally born by Arthur Schuchter at the University of Tromsø (UiT) in 2017 and then realized in Austria later that year. At the club, basics in programming and in engineering subjects are being taught to students of all ages and various skill levels. All the courses are free and are aimed to increase the interest in IT and STEM in general. Knowledge should be free because we firmly believe access to knowledge should always be free and therefore accessible to everybody [1].

We wanted to stay on the pulse of the software development industry so that we can better prepare our students for the rapidly changing job market in the technology sector. IT is used in many areas and skilled workers are sorely needed, but many people are afraid of studies in IT because they are regarded as difficult. However, many of the competences can be learnt over a short period of time, even by people with hardly any background knowledge in this field. There is no need for every applicant to an IT job to have completed a bachelor's or a master's degree. An increasing number of companies such as Google, Facebook and Amazon hire people based on their skills and not only relying on their education. Many of these people have not completed a study in IT. We believe that a lot of tasks in IT jobs can be compared to craftsmanship what is something that can be learnt.

We use the coding clubs to spark young children's interest in IT, to spark interest in University students and we use it as a recruiting tool for prospective universities. We also use Coding Club courses, where we are focusing on teaching the participants skills in order to enable them to get a job in the engineering field. It was a cooperation between the University of Applied Sciences Salzburg and the University of Tromsø (UiT) with the aim to raise interest in IT and to teach students advanced programming. The entire project also aims at arousing girls' interest in IT topics and to the creative building processes. With our approach, we have shown that students of all ages, backgrounds and genders can be fascinated by coding and that their learning outcome is greater when theoretical and practical approaches are combined [1]. In addition, the unemployment service in Austria established a program called FiT (Frauen in Handwerk und Technik). This program is especially designed for women who are seeking to work in a profession that combines craftsmanship and technical skills. Coding Club offers workshops especially for women to help them overcome the barrier of coding. Specialized training will help them to become a professional. The biggest benefit of the FiT program is that women will be offered jobs with a higher salary because of their newly acquired coding skills. This will have a positive impact on the skilled labor shortage in IT [1].

A more recent idea was to include making and coding as two sides of the same coin in order to show the relevance of craftsmanship and creating hardware as opposed to solely focusing on software and programming. We live in a world that often focuses on software, and even data scientists nowadays have little understanding of hardware problems, memory storage issues or inter-device communication. We want to show people that hardware and software go together.

One of our goals is to generate a better understanding and a greater learning curve in programming by involving the students in all stages of product development. Hence, the Coding Club came up with the idea of drones (see Fig. 1). It was our goal to combine hardware and software components to build a system that is based on hardware but can be controlled by software. By also using hardware components, attention was drawn to the making process of devices and the specific problems. It entails such as memory allocation, memory transfer or interfaces. Our experience showed that the inclusion of hardware and the whole building process of drones was perfectly suited to shift the focus to real hardware problems [8].

Fig. 1 Drone workshops at the university

Learning to code seems like a difficult process, but it is in fact rather simple and akin to taking on a new language. Coding is at the heart of all of today's technology. Today, every child needs English, because it is the global language of communication. Similarly, coding could be seen as a technological language which no child can be without it in the future. On a basic level, coding is learning to communicate with computers. We use it to run all the apps, video games, websites, and daily interactive digital experiences. A few years ago, coding was simply something out there and a novelty in educational settings.

Today, we know that around 70% of all new jobs will be in computer sciences. This calls for a new outlook on teaching how to code. As it is in any other language, the earlier you grasp its construct, the easier it is to learn and improve your skills. This means that teaching children how to code should start at a young age in order to make it easier for the students to develop programming skills.

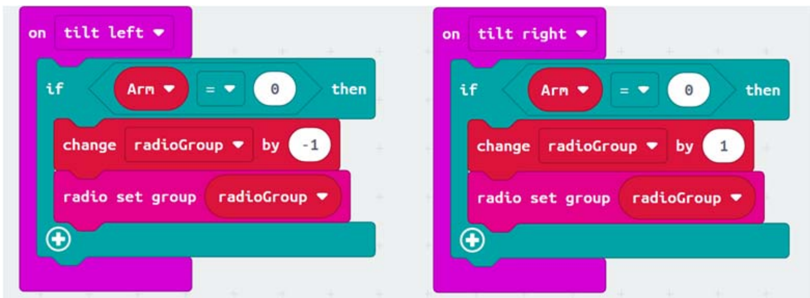

Fig. 2 Micro:bit Programming code example

What is amazing about coding and programming is that it involves communication, problem solving, collaboration and creativity (see Fig. 1). Learning a programming language gives students the knowledge to better make sense the world around them. It makes it easier to understand how our smartphones, laptops, social media and video games work. Basic coding and programming knowledge will therefore change the way learners interact with these technologies.

IV. Co-Creative Educational Workshops

We present two different approaches for our co-creational workshops. First, we concentrate on how traditional engineering fields such as mathematics and physics are being taught online by using flipped classroom problem-based learning environments, where we put the students into the role of a co-educator and co-generator. The second approach deals with project-based approaches over a longer period of time focusing on the combination of making and engineering fields using drone-building. Assembling, screwing, and building drones is perfect for sparking interest. Other theories such as 3D modeling, coding and mechatronics can be used as thematic anchors [8].

The significance of online education has increased tremendously due to the pandemic situation. It is not just a helpful tool for teachers anymore - it is inevitable. The use of videoconferencing systems makes it easy for people to connect, in a private as well as an educational context.

Using online learning tools is a constant learning process not only for primary school pupils but also for university students. Nevertheless, it is crucial that we take the time to use these tools, detect possible problems and deal with them in order to improve future learning experiences. This is due to the fact that we want to be able to cope with another pandemic situation. Therefore, Coding Club has carried out several online-based workshops already.

A. Co-Creative and Co-Educational Online Education to Teach Engineering Subjects

First, we create shared documents where the students and the educator create contents during the course of every module. Students co-create solved solutions, handbooks and FAQs focusing on coding, mathematics and physics [15].

By co-writing we are able to include social aspects and enable the participants to develop team-building skills. Before students are allowed to work on their problems, they are advised to write down their milestones as well as their thoughts into the shared document. They are urged to write everything down in full sentences, in order to avoid wrong implications, to help other students understand what they are dealing with at the moment and therefore enabling direct feedback and interaction with improved motivation [12].

Goals and solutions are split up into processes, and ways of solving problems. Thus, the whole problem-solving pipeline is visible to others and enables better intra- and inter-group collaboration and communication. To write, format, generate and work in a shared document means, you have to formulate it in a way that others understand which in turn helps you to enhance your own knowledge [7].

Second, we put the students into situated learning and flip the problem-solving pipeline using problem-based learning. We conduct the classes online and collaboratively use all the ideas to “reverse engineering” the work towards a goal or the problem to find various ways of achieving solutions. This reversible way of flipping the pipeline enables a more creative and innovative perspective to problem solving [7]. Especially for younger students providing the usage of various knowledge transfer methods (writing, audio, video etc.) is necessary to accomplish interesting and exciting education. Therefore, we do not only use documents, we also use shared whiteboards to discuss ideas. Then we draw together to enable more creative ways of handling problems and use our own handwriting and generate posters as a team (online) as an extra impulse [9], [10].

This study shows the enhancement of a novel for the customary flipped classroom technique. It adds components to the learners, the role of the instructor, the peer assessment as well as the implemented credentialing, which are emphasized. It also points out the important use of digital tools and using technology, for both settings (in and outside the classroom). The students learn this way in a collaborative setting environment, where content acquiring and co-education takes place. Individual and collective learning is enhanced as well as self-regulated strategies, for in and outside the classroom, are introduced. The content co-creation part and the summary of the studies are outlined. In the virtual team setting the aspects of self-regulation as well as co-regulation were condensed. The feedback for using technology, in and outside the classroom, was also positive [2], [3], [9], [10].

B. Co-Creative Workshops to Teach Engineering Subjects Focusing on Drone-Building

We use drones as an anchor to combine hardware and software aspects and to teach coding in engineering fields. The core of the drone is the BBC Micro:bit, a single-plate computer for educational purposes, which offers numerous teaching possibilities from block-based graphical programming to professional scripting languages like JavaScript or Python (see Fig. 3).

The best way to learn is to be fully engaged. This will work best with cooperation and utilization. With creating an Air:bit drone, students will not only use their hands, they have to work collaboratively and change from being a digital consumer to an actual creator. They are placed into the role of a co-content generator and co-educator. Learning with this hand-on approach will help to retain the gained knowledge. It is a fantastic way to use technology in order to create a more multi-sensory classroom, based on communication, innovation, and collaboration skills [16]-[18].

By understanding and working with the hardware when building the drone as well as controlling it with software they have written themselves, we encourage logically structured thinking in our students. We follow a problem-based learning approach, meaning that students are given a specific problem (i.e., to make the drone fly) and then have to come up with solutions on their own [19]-[22].

Our focus lies on a self-determined, self-discovered and hands-on approach for students to develop new skills intuitively and efficiently. While working on this project, students had a chance to encounter real-life usage with practical use. This way they were able to work through all the different stages of creating a commodity [5].

Things that fly have fascinated people for centuries. That is the reason why we chose drones as our object of study. The concept of flying vehicles has always been exciting for people of all age groups, but especially kids. If the flying object, in our case a drone, is something that you built yourself, a feeling of pride and accomplishment to completing such a difficult task is an add on and it also gives you the opportunity to show at the end your finished product of your studies to others. You have not only learnt something; you also have something to show for it [7], [11], [13], [23].

With this ambitious project, we have chosen a difficult path since it is our goal to teach programming to a diverse target group. We are striving to teach different age groups with the same model, from elementary students to university level. We are trying to support students in their learning process by setting up virtual classrooms in their powerful Web IDE.

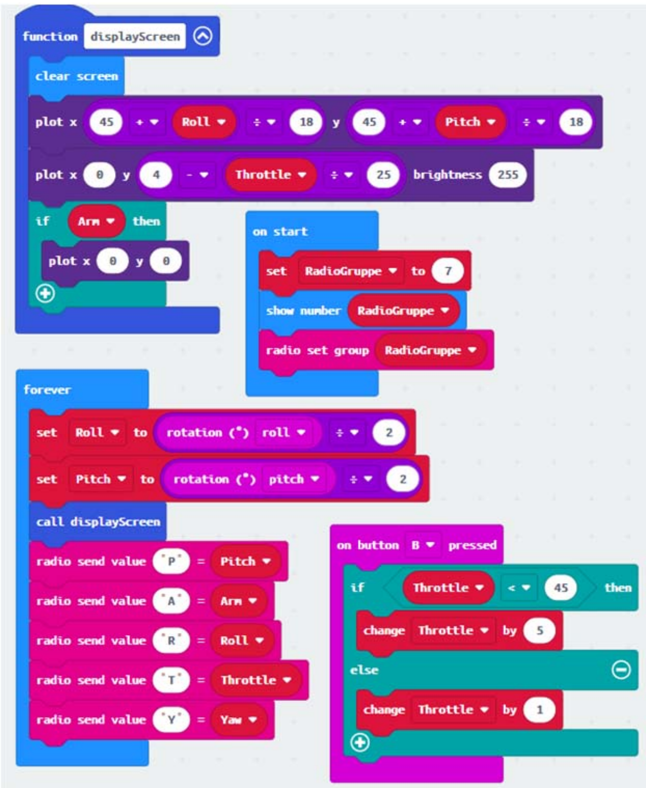

Fig. 3 (a) Micro:bit Block-programming

```
function displayScreen () {
    basic.clearScreen()
    led.plot((45 + Roll) / 18, (45 + Pitch) / 18)
    led.plotBrightness(0, 4 - Throttle / 25, 255)
    if (Arm) {
        led.plot(0, 0)
    }
}
function yaw () {
    pins.analogWritePin(AnalogPin.P1, 1023)
    Yaw = 0
    if (pins.analogReadPin(AnalogPin.P2) > 500) {
        Yaw = 30
        led.plot(3, 0)
    } else if (pins.analogReadPin(AnalogPin.P0) > 500) {
        Yaw = -30
        led.plot(1, 0)
    } else {
        led.plot(2, 0)
    }
}
input.onButtonPressed(Button.A, function () {
    if (Throttle < 45) {
        Throttle += -5
    } else {
        Throttle += -1
    }
})
```

Fig. 3 (b) Javascript code example

The different difficulty levels of the various tasks gives every student the opportunity to have a feeling of success (based on the ARCS model for e-learning by Keller/Kopp), which will benefit their learning process. Moreover, we teach self-regulated learning to consolidate knowledge and to accommodate individual learning methods and learning pace [2], [3].

First, we give the students an already assembled drone. Their first task is to write down an inventory list of all the parts they see without disassembling the drone. As a next step we discuss how the parts (drone and remote control, for instance) could communicate with each other and discuss the Micro:bit on the drone as well as the motors and the propellers. At this point we take some time to delve into the basics of physics and mathematics. Second, we let the students reassemble the drones while they are urged to write down the process in reverse order.

Information and special hints that can be useful for everybody are written into a shared FAQ document to help other teams, if needed, and to generate a document, which can be used in the future. Especially propellers and motors and the coding of the drone gives us many possibilities to create a useful document, because working with hardware as well as coding is often like an error searching process, where a handbook of problems is regarded as very useful [12], [14], [16].

Our experience shows that a shared document is also very useful when working on higher level coding problems (i.e., including python programming with the drones, including camera and machine learning algorithms with camera data).

What is important for us is that the participants gain knowledge. That is why we explain physical elements by using 3D modeling and theoretical aspects of physics such as thrust or rotation by using propellers or motors of the drones. We use mathematical knowledge in 3D modeling (shaders) and always combine those theories with the drone's assembly. Furthermore, we believe teaching mechanical elements like tools as well as coding is inevitable, since both are equally important when building a drone [5], [13], [23].

Students start with an easy programming language and with block-based programming. They can then improve their knowledge by using a different coding language such as JavaScript or Python, by using additional devices such as cameras or sensors to create embedded systems. Once they have established a basic knowledge of the subject, they can then explore different options. Many different fields of engineering subjects are possible by using the drones as a hook to spark interest and to transfer knowledge.

To give a better feedback to the students we make use of the Micro:Bit's classroom tool during the workshops. This tool enables the teacher to see their students' code in real time, to select and release their own code and/or to all students, and to interactively manage all codes on their own computer without having to check their students' computer screens (classroom.microbit.org).

If we use the workshop series as coding clubs for the general public, we set up the workshops with an online registration, and ticket system. Then, the drone packages are dispatched, followed by an online zoom meeting of all participants. Individual coaching is given, and there are zoom polls to check on students' progress.

V. Experiments and Results

We used four different test groups to show if the students could gain knowledge as well as team-player skills during our sessions. Test group 1 consisted of 16 first year university students who were put into the role of content generators and co-educators. They were tested based on their shared compendium, which they created in groups of four and no other material was used. Test group 2 consisted of 15 first year students who were using books for mathematics, physics and programming without using a shared compendium. Test group 3 consisted of 16 high school students with no experience in programming, hardware nor drone-building working in four groups with four persons working with one drone. Test group 4 consisted of 10 people working individually on a drone not interacting with other people.

As a first step, we wanted to find out if knowledge could be transferred better with a shared compendium or by using traditional learning materials. Both groups were lectured online, test group 1 was ordered to create a document with all the theories they learned during the lectures, new knowledge was only taught by flipped classroom problem based assignments. In some occasions, solutions and goals were split up and discussed and reversed to better understand the components of the problem-solving pipeline. Test group 2 was also taught online with lecture notes and additional books for engineering from the beginning on.

Test group 1 and test group 2 were compared if co-content generation could improve knowledge transfer. We applied pre- and post-lecture tests to examine mathematical, physical and coding knowledge transfer. Both pre- and post-lecture tests consisted of the same 20 questions, only the order was changed in the post-lecture test. Fifteen sessions with each group were applied and the results were averaged. The test scores ranged from zero (worst) to 20 points (best).

Results are given in Table I. It can be observed that the group using a shared compendium could achieve a higher amount of correct answers at the post lecture tests; especially the number of correct answers concerning coding-relevant questions by nearly 10 points. We believe learning to code in teams in a collaborative way could lead to a better output as opposed to working as an individual. In addition, a third-year student was asked to undertake a satisfaction-level survey after the sessions by using numbers from 1 to 5 (5 is meant as the full satisfactory level). The overall satisfactory level has increased by 0.5-1 points.

TABLE I
Comparison between Co-Content Creative Lectures and "Traditional" Online Lectures

	T1: pre			T1: post			T2: pre			T2: post		
Avg. points	M	PH	C	M	PH	C	M	PH	C	M	PH	C
	3.6	2.8	2.8	17.6	14.8	17.2	3.9	2.9	2.8	12.6	13.6	10.3

In another test scenario, we wanted to show if we could use drones as an anchor to teach 3D modeling, engineering subjects and coding. As a first step, test group 3 and test group 4 focused on 3D modeling of drone parts such as propellers and the chassis of the drone. Test group 3 was modeling in teams at different computers interacting with each other, whereas test group 4 was modeling individually. Modeling knowledge was given by the instructor and the results of the models were compared. The resulting models were evaluated by two 3D artists depending on precision, with scores from 0 (worst) to 5 points (best). Table III shows the averaged results of the scores. It can be observed that the overall outcome score is increased by approximately 0.5 points.

TABLE II
Student Satisfaction Feedback

	T1: co-creative flipped classroom			T2: "traditional" education		
Satisfaction survey	M	PH	C	M	PH	C
	4.2	3.6	4.2	4.0	3.2	3.3

TABLE III
3D Modeling Comparison between Team and Individual Modeling

	T3: 3D modeling in groups			T4: 3D modeling individual		
Score (averaged)	Accuracy	Edges	Stability	Accuracy	Edges	Stability
	3.2	3.4	4.2	2.8	3.4	3.9

As a next step, test group 3 and test group 4 were examined concerning engineering knowledge and coding. Before and after a session a test was given with 20 questions each. Table IV exhibits the averaged results and shows by working collaboratively the coding score can be increased by approximately 1.5 scoring points.

TABLE IV
Comparison between Collaborative and Individual Learning

	T3: pre			T3: post			T4: pre			T4: post		
Avg. points	M	PH	C	M	PH	C	M	PH	C	M	PH	C
	5.8	4.3	3.3	12.3	10.4	14.4	5.9	4.2	3.4	12.5	10.2	12.3

As a final experiment, a third-year student was asked again to undertake a satisfaction-level survey after the sessions by using numbers from 1-5 (5 is meant as the full satisfactory level). Results can be seen in Table V. It can be observed that satisfactory level for the test group 3 and test group 4 were similar.

TABLE V
Student Satisfaction Feedback

	T1: co-creative flipped classroom			T2: "traditional" education		
Satisfaction survey	M	PH	C	M	PH	C
	4.3	3.6	4.2	4.4	3.6	4.1

VI. Conclusion

Knowledge in technical engineering fields such as math, physics and IT are necessary in our lives and there is no way we can escape that, whether we like or not. It is therefore crucial for future success on the job market to learn engineering and coding skills from an early age on. In most scenarios, we need an engineer who is a team player, can think creatively, reverse-engineer problems and teach others. The IT industry needs people, who are able to work with hardware and software respectively to yield higher scale applications.

We believe that students who learn to develop social skills such as co-working and are trained in hardware and software will be better prepared to become IT professionals. Programming is one of the most promising fields to delve into nowadays. Learning approaches tend to be more software-based and can quickly lose their connection to real-life applications. This paper proposes a different, more hands-on approach to this topic, using programmable drones as a showcase for engineering subjects and to re-design the traditional flipped classroom approach by giving the students a co-content creator and co-educator role [9]-[11], [13].

Diving deeper into real-life applications is crucial for students in order to understand the relevance and potential uses of what they are learning. In addition to theoretical teaching approaches, which are an important foundation, the hands-on approach will result in a more intuitive and efficient learning experience. It is a promising approach with a wide range of applications that should be included in more primary school curricula. What makes this approach so unique is, firstly, we use the same model in a playful way with blocks as well as properly used pure script language. We can show students how programming comprises both, software and hardware aspects, and they can see the results immediately. We show that programming does not exist exclusively in a software environment. Another vital aspect of our teaching model is that, we guarantee that students feel a sense of achievement, which, in turn, increases their interest and their motivation for IT [5].

It is too late to have your first encounter with collaboration in teams as well as coding at university level. If the combination of making and coding could be implemented earlier, chances are that the output for the IT sector would be much greater. We have to start earlier to introduce students to creative mindsets, innovative solution thinking, reverse thinking, flipping the way we see how solutions and goals are yielded, changing the order of the pipeline and to the approach that making mistakes is okay [9], [10].

In order to prepare students for future careers in IT, we have to adapt and modernize the way that engineering and coding is taught. We have to start working in teams early and support creative thinking and computational thinking at an early stage. We have presented different educational models to teach programming and IT competences depending on the target group i.e., different school types and university settings. We focus on practical, hands-on work such as screwing parts together, working with a user manual, physics, mathematics and IT, approaching problems in reverse order and controlling them. This focus stems from the fact that young people often grow up without a practical, physical connection to things such as a screwdrivers or tools in general, also because they do not see their parents using them.

This paper proposes a didactical approach to teach computational thinking to pupils and students by using drones. The following research questions have been considered: 1) Is it possible to put the student into a co-educator creator role and at the same time establish knowledge transfer? 2) Is it possible to use a hardware-oriented setting (i.e., drone building, screwing parts, flipped classroom environment, active learning) to teach programming paradigms? The authors are currently collecting data to evaluate the process to determine whether the proposed approach is suitable to teach primary school kids (with parental guidance) pupils and students CT and programming as well. Working on a topic collaboratively is also an important aspect of mathematics. "Frontal teaching" can be used for giving inputs, but problem-based learning and the flipped classroom model are even more effective [9]-[11], [13]. However, it is obvious that not every team player wants to work in a team all the time. Therefore, we suggest including a flipped classroom approach for about 50% of the teaching time. Only if profound basic knowledge and motivation can be found in the students, a co-creational flipped learning method can be implemented [5], [9], [10].

As mentioned earlier in this paper, critical thinking, which is a part of operative programming and computer science, has become an even more essential skill for the 21st century. Even experts from all over the world started to think on how to display terminologies like solutions and debugging. One of their first steps was to make the act of coding more accessible to younger and more diverse learners.

It is our aim to arouse curiosity and awaken interest in STEM subjects, especially in IT, to encourage logically structured thinking and to teach problem-solving skills. Experiments show that by using co-creative problem-based educational models, overall knowledge transfer can be increased, and team-building skills are improved. Aiming at equipping students for the increasingly more challenging modern job market, we strongly suggest incorporating co-creative problem-based educational models from an early age. The earlier students are exposed to engineering and collaborative tasks, the more hardware and software skills are combined, the more qualified they will be and the better prepared they will be for their future [1], [7].

References

[1] Ntemngwa, C. & Oliver, J.S. (2018). The Implementation of Integrated Science Technology, Engineering and Mathematics (STEM) Instruction using Robotics in the Middle School Science Classroom. International Journal of Education in Mathematics, Science and Technology (IJEMST), 6(1), 12-40.

[2] Li, K., & Keller, J. (2018). Use of the ARCS model in education: A literature review. Computer & Education, 122, 54-62.

[3] Keller, J. (2000). How to integrate learner motivation planning into lesson planning: The ARCS model approach, Florida State University, February 2000.

[4] Keller, J. M. (2010). Motivational design for learning and performance: The ARCS model approach. New York: Springer.

[5] Keller, J. (1983). Motivational design of instruction. In C. Reigeluth (ed.), Instructional design theories and models. An overview of their current status. Hillsdale, NJ: Erlbaum. S. 383–434.

[6] Keller, J.M. & Kopp, T.W. (1987). An application of the ARCS model of motivational design. In Reigeluth, C. M. (Hg.): Instructional theories in action. Lessons illustrating selected theories and models. Hillsdale, NJ: Erlbaum. S. 289–320.

[7] Schmuck, T. H., Haber, P., & Mayr, M. (2018). Natural science problem-based learning through pupil project-driven sensor environment. In Edulearn 18. 10th International Conference on Education and New Learning Technology (Palma, 2nd-4th of July (2018): conference proceedings (pp. 6522-6527). IATED Academy.

[8] Krajcik, J. & Blumenfeld P. (2005). "Project-based learning". The Cambridge Handbook of the Learning Sciences (Editor R.K. Sawyer), pp. 317-333, Cambridge: Cambridge University Press.

[9] Tugun, V. & Uzunboylu, H. & Ozdamli, F. (2017). Coding Education in a Flipped Classroom. Technology Education Management Informatics. 6. 599-606. 10.18421/TEM63-23.

[10] Strelan, P., Osborn, A. & Palmer, E. (2020). The flipped classroom: A meta-analysis of effects on student performance across disciplines and education levels. Educational Research Review. 30. 100314. 10.1016/j.edurev.2020.100314.

[11] Bartholomew, J.L., & Mayo, R. S. (2018). Development of a 4th-8th Grade Curriculum for Flying and Programming Mini Drones. All Graduate Plan B and other Reports. 1203. UtahState University.

[12] Heinonen, K., De Grez, N., Hämäläinen, R., De Wever, B., & Van der Meijs, S., (2020). Scripting as a pedagogical method to guide collaborative writing: university students' reflections. Research and Practice in Technology Enhanced Learning volume 15, Article nr: 15.

[13] Pharr M. & Wenzel J. (2016). Physically Based Rendering: From Theory to Implementation (Third Edition).

[14] Arvaja, M. & Häkkinen, P. (2010). Social Aspects of Collaborative Learning International Encyclopedia of Education (Third Edition).

[15] Akhilesh, K. B. (2017). Co-Creation and Learning. In: Co-Creation and Learning. SpringerBriefs in Business. Springer, New Delhi.

[16] Chemi T. & Krogh L. (2017). Co-Creation in Higher Education. Students and Educators Preparing Creatively and Collaboratively to the Challenge of the Future. SensePublishers, Rotterdam.

[17] Root, D. (2018). Creating a Culture of Collaboration: 5 Strategies To Help You Do So. eaglesflight, blog.

[18] Bovill, C. (2020). Co-creation in learning and teaching: the case for a whole-class approach in higher education. High Educ 79, 1023–1037.

[19] Jonassen D.H. & Hung W. (2012). Problem-Based Learning. In: Seel N.M. (eds) Encyclopedia of the Sciences of Learning. Springer, Boston, MA.

[20] Wood D. F. (2003). Problem based learning. BMJ (Clinical research ed.), 326(7384), 328–330.

[21] (9) Gallagher, S., Stepien, W. & Rosenthal, H. (1992). The Effects of Problem-Based Learning on Problem Solving. Gifted Child Quarterly - GIFTED CHILD QUART. 36. 195-200.

[22] Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., & Ryan, M. (2003). Problem-Based Learning Meets Case-Based Reasoning in the Middle-School Science Classroom: Putting Learning by DesignTM Into Practice. Journal of the Learning Sciences, 12(4), 495–547.

[23] Carle A. & Schertle R. (2017). Make: mBot for Makers - Conceive, Construct and Code Your Own Robots at Home or in the Classroom. CA, United States: Maker Media, Inc.