Looi, C.K., Wadhwa, B., Dagiené, V., Seow, P., Kee, Y.H., & Wu, L.K. (Eds.). (2021). *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021*. Singapore: National Institute of Education.

A Framework for Integrating Computational and Design Thinking Processes

Riccardo CHIANELLA[1]*, Diego REITANO[2], Ettore MORDENTI[3], George BARITSCH[4]
[1,2,3,4] School of Design, Politecnico di Milano, Italy
riccardo.chianella@asp-poli.it, diego.reitano@asp-poli.it, ettore.mordenti@mail.polimi.it, georgeoscar.baritsch@mail.polimi.it

ABSTRACT

Due to the rapid change brought by new emerging technologies, computational thinking (CT) has become a fundamental skill. Contrarily to the large number of studies focused on introducing CT in STEM subjects, we direct our research towards a broader context, that of design. Given the importance of CT concept acquisition in terms of future design thinking education, this paper presents a qualitative study at the intersection of teaching design thinking and CT. We develop an innovative framework to integrate the two processes in design courses and we explore its potential and limitations with design lecturers who could potentially introduce the framework in their teaching practice. Moreover, we reflect on what needs to change for CT education to be successfully implemented in design schools across the world. This study refers to the example of Italy which, similarly to other countries, could constructively improve its design teaching with CT to secure its large design industry for the future.

KEYWORDS
computational thinking, design thinking, design university

1. INTRODUCTION

1.1. Context

Technological progress directly impacts the emergence of new skills required by workers. The integration of these skills should be at the center of attention for those educational institutions preparing students entering the job market with relevant courses and subjects. In particular, the fourth industrial revolution gave life to fast-moving technological trajectories enabling new forms of creation based on the development of augmented, ubiquitous and embedded technologies, where computation sits at the core of the design production (Schwab, 2017). Therefore, we argue that computational thinking (*CT*) should be integrated into design to support its rapid evolution in the technology era. By teaching designers how computers think and integrating it within their practice, they can better cope with emerging technologies. CT prepares students to become better problem solvers and critical thinkers (DeSchryver & Yadav, 2015).

Existing research successfully explored possible ways of introducing computational thinking concepts in university non-STEM subjects. For example, Basawapatna et al. (2011) applied the CT process to game design. However, existing research in the design field still considered CT only as a hard skill, merely linked to coding or 3D modeling. Given the lack of studies considering CT as an integral to the design process, we identify a research gap in the field of CT for design education. This work aims to fulfill this gap by proposing a framework implementing CT into design thinking which could be applied to a broad variety of design classes.

This paper is organized as follows: first, we present the affinities between design thinking and CT. Subsequently, we introduce and define a proposed framework combining the two processes. Through a series of interviews, we test how the framework could be implemented into real-life design studios and workshops. The resulting findings will lead into a discussion which aims to identify its positive aspects and limitations. We finally describe the further research that has to be developed in order to better integrate CT in design thinking.

1.2. Affinities between CT and design

To ensure coherence throughout the article, we adopt Wing's definition considering CT as "an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing" (Wing, 2008). According to Wing, learning CT concepts is now seen as a practice for leading students to develop more transversal skills which do not just include programming. As reported by Soleimani (2019), computation should be considered as a thinking process, as "it is about effectively structuring information and developing logics". Tabesh (2017) proposed a four-stage model of the computational thinking process: decomposition, pattern recognition, abstraction, algorithm design. Following these premises, we argue that the implementation of CT in design education should integrate the processes of design, rather than the tools.

In one of his writings, Denning analyzed the potential in combining CT and design thinking. He stated that "If the two kinds of thinking were blended together, some significant advances in software design and development would surely follow" (Denning, 2013). Moreover, Shute, Sun & Asbell-Clarke defended that CT could help designers go beyond the limits of design thinking, which is still too tied on "product specifications and the requirements imposed by both the human and the environment" (Shute, Sun, Asbell-Clarke, 2017). These statements lay the foundation to our proposal.

The way in which design is taught among most universities around the world is by giving value to the development of a design process. This plays a crucial role in guiding designers across projects, whether they are designing objects, clothing, interfaces or interiors. Similarly to CT, the design process cannot be simplified into a problem-solving activity (Goel & Pirolli, 1992), yet it is still based on an iterative and step-by-step sequence of actions (Lawson, 2006). Many design processes have been created, each one with a specific focus, content, structure or graphical notation (Bobbe, Krzywinski & Woelfel, 2016). Despite that, all processes show many similarities (Eckert & Clarkson, 2005).

For this study, we focused on the widely-known design thinking process developed by Stanford d.school (Plattner et al., 2009). This process integrates most of the existing ones;

it is taught in many design universities and has been promoted by numerous companies from the design field, including Apple, IDEO and SAP (Efeoglu et al., 2013).

1.3. Framework

Our proposed framework associates the four stages of CT (decomposition, pattern recognition, abstraction and algorithm creation) defined by Tabesh (2017) to the five stages of Design Thinking (empathize, define, ideate, prototype and test) described by Plattner et al. (2009).

Previous research has proven that the best way to teach students about CT is by associating its basic principles to already-known practices within their subject (Lu & Fletcher's, 2009). Moreover, the framework is shown in a circular ring exemplifying it as a process that never ends. The proposed framework is visualized at *Fig.1*, followed by a description of how stages are linked to each other.

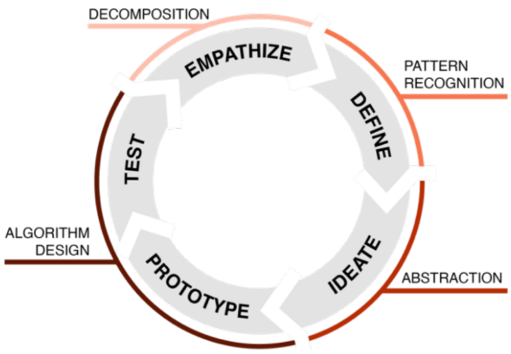

*Figure 1. "*Circular Framework For Computational And Design Thinking Processes". Design Thinking process (inner circle) and CT process (outer circle).

- **Empathize**: it is the stage in which designers come to understand people who experience a certain need or problem through ethnographic (or desk) research and observation. Through **decomposition** (CT), designers deconstruct a problem in many parts.

- **Define**: designers analyze deconstructed information and use the CT principle of **pattern recognition** to formulate insights: non-obvious, actionable statements that show a deep understanding of the investigated problem or need. They lead to a design challenge.

- **Ideate**: the phase in which the challenge is taken on and multiple solutions are generated to address it. A degree of openness to outer influences is required in order to produce innovative ideas. **Abstraction** is the CT principle that lets designers expand the solution space into other contexts and ultimately find a wider range of solutions.

- **Prototype and Test**: through prototyping designers produce artifacts that represent the solution in its current status. By testing the solution, designers evaluate it and identify areas of improvement. Subsequently, the whole process is iterated until the final solution is implemented, following a process of **algorithm design**.

In order for students to fully take advantage of the framework, they must put it into practice in their design studios. Following Volcz (2018), we suggest alternating a series of 4 to 5 theoretical lectures, one for each stage of the framework, with hands-on practice, which can be achieved in design studios or workshops. Directions on the specific deliverables should be defined according to the specific type of project. We will do it by applying it to three exemplary design classes.

2. METHODOLOGY

To further analyze how the framework could be put into practice in a real design course, interviews were conducted with three prominent lectures of the School of Design of Politecnico di Milano. We kept our research qualitative rather than quantitative to collect in-depth insights tied to the specific needs of each design course taught by each interviewee. These interviews were semi-structured and consisted of a set of priorly-defined open-ended questions. They took place via an institutional teleconferencing platform and lasted about one hour each. Interviews began by questioning interviewees' previous knowledge and CT understanding in relation to Wing's views. After that, lecturers were shown the framework we developed. The discussion was structured by analyzing each phase individually, asking the interviewees to comment on its consistency and applicability in the didactic context, concentrating on their research area. In regard to the latter comment we asked interviewees to also provide real examples in order to make the modus operandi more understandable.

Unsure on their level of CT understanding, we shared an introduction to our study with the participants prior to our interview. We ensured that all our participants knew that we were referring to Wing's definition of CT and we gave them the chance to ask for clarifications. Therefore, this section was a chance for making sure that participants understood the purpose of our study, thus ensuring valuable feedback. After our introduction, interviewees focused on the framework. More specifically, they shared their general impressions on how the CT process could enrich their students' learning outcomes. Then they went through each individual step of the framework and theorized some possible hands-on assignments based on the current courses that they taught. Finally, they shared eventual perplexities or improvements to the framework. Participants were asked to think out loud, allowing us to follow their reasoning and ensure that their suggestions were reliable and logical.

Collected data was analyzed through affinity diagramming to organize what could seem unstructured or dissimilar qualitative data (Hartson & Pyla, 2012). Contextual inquiry data containing quotes from the interviews was fragmented into post-its. Then the post-its were clustered according to their similarities. Finally, clusters were atomized into concise insights.

3. FINDINGS

The table below (*Table 1*) reports the results of the interviews regarding some practical ways in which the Circular Framework for Computational and Design Thinking Processes (*Fig. 1)* could be implemented in a design course.

Table 1: Application of the framework to some design courses

	Course in: Knitwear Design	**Course in: Shapes and Algorithms for Generative Design**	**Course in: Methods and Instruments for Design**
Empathize (Decomposition)	Study how fast animals run. Clusterize findings based on movement, anatomy, species, etc.	Study a particular natural phenomenon by breaking it down into its constituent elements. ***e.g.*** Analyze waves in liquids.	Study the structure and behavior of resistant and light materials. Break down findings into clusters.
Define (Pattern Recognition)	Within clusters, find a pattern that provides the key to solving the design challenge. ***e.g.*** the shape of the paw or texture of the skin.	Pick inspiring behaviors and find the pattern that makes them similar. ***e.g.*** When objects fall in water, they create ripples.	Identify cross-cluster patterns that provide a solution to a certain problem. ***e.g.*** Which structure is the lightest and most rigid one?
Ideate (Abstraction)	Abstract the findings and create a concept. ***e.g.*** The texture of the skin inspires a new material for a shoe.	Abstract that pattern and create a codethat resembles it. ***e.g.*** An input for a 2D visual effect that resembles ripples of water.	Abstract findings and integrate the structure into an existing object. ***e.g.*** The structure can substitute plastic parts in safety helmets.
Prototype and Test (Algorithm Design)	Make prototypes and incrementally improve the required features. Test and reiterate. ***e.g.*** When tested, does this correlate with improvements in running?	Complete code. Run it and test it. Improve it and iterate the process till the desired effect is achieved.	Write a code that recreates the chosen 3D structure (CAD modeling, mathematical strength tests). 3D-print the structure and test it in real life scenarios.

4. DISCUSSION

4.1. Positive aspects of the framework

As reported by the interviewed lecturers, the mathematical and programming skills of design students in Politecnico di Milano are perceived as relatively low when compared to the European standards. Participants shared the common belief that this was due to the students' low interest in these subjects and a lack of depth being offered in these areas. It was thus important to create an accessible frameworkfor students without an advanced knowledge in math orcoding.

"By looking at the background our students have, this [framework] is the only way in which design students could ever understand it: by comparing it to their reality." (Lecturer in Methods and Instruments for Design)

Design lecturers referring to their previous experiences generated another powerful finding: by integrating CT in the design process, students learn how to be more versatile. For example, a participant reported the example of a shoe design project, where students incorporated computational tools in the "empathize" stage. He mentioned the act of generating new insights by studying the aerodynamics of distinct solid shapes instead of referring to existing solutions. By quoting the lecturer:

"Once implemented in design processes, students who know computational thinking will be able to ask the right questions about their projects. They will learn how to go beyond the study of existing products by abstracting the modalities with which a product is used." (Lecturer in Knitting Design)

Moreover, lecturers perceived it as being particularly suitable for advanced design courses and as a research tool where students who are already familiar with the design process can confidently change their workflow and apply CT to design innovative projects.

"To get the full potential out of this framework, I would rather introduce it in the Master's course I'm teaching, rather than the Bachelor's one." (Lecturer in Shapes and Algorithms for Generative Design)

"This framework is more linked to our research fields, as it allows a deeper understanding of the topic." (Lecturer in Methods and Instruments for Design)

4.2. Limitations of the framework

During the interviews, we came across different definitions of CT introduced in the examples given to us by the lectures, causing time to be spent to ensure a common understanding. Additional efforts must be put into establishing a shared understanding, prior to CT being introduced into design classrooms within an institution.

Another limitation recognizedby the interviewees regarding the proposed framework concerns the fact that our association of CT and design thinking works only at an introductory level and cannot be applied to learn topics too far from design. The idea of using our framework solely in an introductory level is also partially due to a lack of skills of students regarding computation. According to our interviewees a greater knowledge of hard sciences would be needed to be able to fully understand the CT process. However, the introduction of hard science subjects could be a deterrent for students to enroll:

"Students who come to design school do not like math. This implies that if we add more math courses, less students will apply, and we will lose funds. Many students coming from high schools are frightened by these subjects." (Lecturer in Methods and Instruments for Design)

Our interviews supported that for most projects an overlap can be seen between CT and design thinking. However, there seem to be some design areas in which CT should *not* be integrated. This is for areas with a strong sensorial component (*e.g.* fashion design), where a project's success heavily relies on factors that cannot be abstracted, like the sensorial perception of a material to the users. There is a gap between the minimum level of abstraction required by CT and the sensorial qualities of certain design contexts.

"…An important role in the design process is the presence of errors, which can often generate an interesting finding. I believe that by applying CT, some projects would be error-free, and thereby become less innovative." (Lecturer in Knitting Design)

Finally, according to two of the interviewees, the collaboration of design lecturers and computer science lecturers is preferable to develop an effective program to introduce CT as an integration to design thinking. Thislevel of multidisciplinary collaboration has yet to be achieved, though through the interviews it was found to be feasible.

5. CONCLUSIONS

This study contributed to the creation of a new method to introduce CT to design students. This interdisciplinary approach was finalized with the creation of a methodical framework merging computational and design thinking. Although this was designed to be implemented in those design universities where students lack mathematical and programming skills, our study focused on the School of Design of Politecnico di Milano. Here, the design process has never been associated with CT before. Our work can hence be considered as an example to address CT in other design schools around the world.

After analyzing the opportunity from a theoretical perspective, we developed the 'Circular Framework for Computational and Design Thinking Processes' with the purpose of integrating CT in the design thinking process. This framework does not see CT as a compulsory skill for designers. However, by introducing it alongside design thinking in traditional design methods, it can help shape designers who are more aware of technological power and are more versatile. The framework was further developed by running some qualitative research with lecturers in the School of Design. Design lecturers were asked to further improve the new method by sharing their expertise and applying it to the courses they were currently teaching.

The findings from our research justified how to introduce CT to design students and shared how it could be used to design innovative projects. However, the model will still suffer when put into some design teachings as one cannot consider the individuality of each project or course. Moreover, lecturers expressed their wish for establishing new collaborations between design and computer science experts to introduce the topic more properly. The framework and definition of CT was discussed though views still contradicted in small areas, exemplifying why design scholars must agree on a single CT definition for this framework to be utilized.

Now that the framework has been defined, we strongly believe that the first step towards a more in-depth version is to test it within a design studio. Recognizing the weight that a student's perception has on reliability and applicability of our framework, further development should include a qualitative collection of students' feedback on the "Circular Framework For Computational And Design Thinking Processes". Moreover, we should include some quantitative research method, for instance by assessing the level of CT skills of students prior to and following introduction to the framework. Finally, given that our research was conducted in Politecnico di Milano, we would like to draw attention on other design schools in other countries to further research on how this framework could be implemented in their curriculum. A global discussion on the topic would bring up new limitations and advantages, hence improving the framework on a world-wide level. The concept stemming from our work is thereby an attempt to stimulate a deeper reflection on the intrinsic relationship between design education and CT. This could be expanded even more by exploring how design thinking practices can be applied to the design of computational solutions.

6. REFERENCES

Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011, March). Recognizing computational thinking patterns. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, 245-250.

Bobbe, T., Krzywinski, J., & Woelfel, C. (2016). A comparison of design process models from academic theory and professional practice. *In DS 84: Proceedings of the DESIGN 2016 14th International Design Conference,* 1205-1214.

Denning, P. J. (2013). Design thinking. *Communications of the ACM*, 56(12), 29-31.

DeSchryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education*, 23(3), 411-431.

Eckert, C. M.; Clarkson, P. J. (2005): The reality of design. In P. J. Clarkson, C. M. Eckert (Eds.): *Design Process Improvement A review of current practice*. London, 1–29.

Efeoglu, A., Møller, C., Sérié, M., & Boer, H. (2013). Design thinking: characteristics and promises. In *Proceedings of 14th International CINet Conference on Business Development and Co-creation* (pp. 241-256).

Goel, V., Pirolli, P. (1992): The structure of design problem spaces. In *Cognitive Science 16* (3), pp. 395–429

Hartson, R., & Pyla, P. S. (2012). The UX Book: Process and guidelines for ensuring a quality user experience. *Elsevier*.

Lawson, B. (2006). How designers think: The design process demystified. *Routledge*.

Lu, J.J., & Fletscher, G.H.L. (2009). Thinking About Computational Thinking. *ACM SIGCSE Bulletin, 41(1), 260-264.*

Plattner, H., Meinel, C., & Weinberg, U. (2009). Design-thinking. *Landsberg am Lech: Mi-Fachverlag*.

Schwab, K. (2017). The fourth industrial revolution. *Currency*.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158.

Soleimani, A. (2019). Computational Design Thinking and Thinking Design Computing. In *2019 Reynolds Symposium: Education by Design. Portland, Oregon, October 18-19, 2019*.

Tabesh, Y. (2017). Computational thinking: A 21st century skill. *Olympiads in Informatics*, 11(2), 65-70.

Volcz, I. (2018). The Use of Computational Thinking to Advance Learning in the Pre-university Subject of Digital Literacies. *In Proceedings of the International Conference on Computational Thinking Education 2018. Hong Kong: The Education University of Hong Kong. 68-71*

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions Of The Royal Society A: Mathematical, Physical And Engineering Sciences, 366(1881), 3717-3725.*