IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol.006, LIIR391
DOI: https://doi.org/10.52731/liir.v006.391

The Taxonomies Project: Coding as Botanical Art

Andrea Valente *, Shanice Otersen †,
Emanuela Marchetti ‡

Abstract

This paper presents Taxonomies, a STEAM project that can be placed on the intersection between art, science, and technology. In Taxonomies, art is taking the lead role, providing a goal for action, while sciences and computation provide a shared agenda and means for artistic creation. The project is a collaboration between the University of Southern Denmark, the Biotech Lab at Spinderihallerne and Rosborg Gymnasium, a Danish secondary school. During the past year a series of 3 workshops was designed, implemented, and run at Rosborg. Here we focus on the challenges, design solutions, and findings related to the third workshop, which aimed at introducing forms of creative, aesthetic coding suitable for art students with no prior experience and allowing them to explore algorithmic botany. Results include a general method to convert L-Systems to tangibles, an intuitive software simulator to create simple algorithmic plants; moreover, materials and methods developed worked well within the strict constraints of the project. A new edition of Taxonomies is being run in these months, and we are working on generalizing the workshop format to a reusable and shareable kit for other schools.

Keywords: algorithmic botany, botanical art, Computational Thinking, STEAM, tangibles.

1 Introduction and Motivation

This project started as a collaboration between the University of Southern Denmark (SDU) and Spinderihallerne, an innovation center located in Vejle (Denmark), dedicated to art, technology, and learning. Spinderihallerne hosts a Fablab equipped with fabrication machines, and a Biotech Lab committed to the research of organic sustainable materials. Researchers from SDU and the director of the Biotechlab teamed up to investigate how art making could be framed as a platform to foster learning of sciences, across schools and non-formal learning environments.

Our study starts from the premise that in the past art and natural sciences were in a symbiotic relationship, as naturalists were themselves engaging in naturalistic illustrations as a way to support their analysis and documentation process. Excellent examples of naturalists who used arts in their scientific studies include: the work of Leonardo in the Renaissance and more recently of Ernst Haeckel and Beatrix Potter before she became a famous author, and currently pencil drawings and copper etching illustrations of microorganisms by Anderson-Tempini [1].

* Department of Design, Media and Educational Science, University of Southern Denmark (Kolding, Denmark), https://orcid.org/0000-0002-6295-9511
† BioTechLab Spinderihallerne, Department of Innovation and Entrepreneurship (Vejle, Denmark), https://orcid.org/0009-0005-6440-3268
‡ Department of Mathematics and Computer Science, University of Southern Denmark, https://orcid.org/0000-0002-7949-9953

Historically botanical art emerged in connection with a fascination for natural forms, moreover, empirical knowledge in natural sciences provided artists with the ability to make their own colors, brushes and medium fluids. On the other hand, knowledge of human perception related to composition, contrast and color enabled artists to create captivating illustrations. On the other hand, the beauty of nature has fascinated mathematicians for centuries [2], in relation to human proportions, and the many forms of plants and flowers. This led the way to computational studies aimed at reproducing botanical forms through code on a computer. In this sense, we find a deep connection between botanical art and sciences, as botanical art has as a main goal to reproduce forms of plants and can leverage knowledge from biology and chemistry to investigate plant forms and to extract organic colors from plants. At the same time, botanical art can leverage knowledge related to mathematics and Computational Thinking (CT for short) to algorithmically create images, revealing the computational complexity behind the beauty of plants (as in [2] and [3], chapter 6).

Therefore, we have investigated in collaboration with Rosborg Gymnasium, a humanities-oriented secondary school in Vejle, a town located in Southern Jutland, Denmark, how to integrate art and science in organic ways, so that art could provide the basis for a new understanding of natural sciences and CT. This investigation started in spring 2023 and resulted in the creation of a series of three workshops aimed at integrating art and sciences; we ran the workshops in spring 2024. The first workshop is aimed at teaching the students to make their own colors out of various plant matter and food waste. During the second workshop, the students were introduced to botanical art as a scientific method to gather and document data from in-depth observation of plants, and were instructed to create botanical illustrations using the colors they made in the previous workshop. Finally, the third workshop should be designed to introduce some form of creative, aesthetic coding, coherent to the main theme and suitable for art students with no prior CT experience. This is the main challenge we will discuss in this paper, and our solution involved algorithmic botany [2] and manual algorithm simulation based on tangibles, to allow students to procedurally generate images representing different forms of grass and plants.

We call our study Taxonomies, with reference to Linnaeus and his categorization of different living species, and the symbiosis we identified between art and sciences in early naturalistic studies. Our study can be placed on the intersection between art, science, and technology, in which art is taking the lead role, providing a goal for action, while science provides a shared agenda and means for artistic creation. A stunning collection of artworks was gained from the workshops, which was displayed at an open summer exhibition at Spinderihallerne in Vejle, where the Fablab is located.

In this paper we focus on the insights we gained from taxonomies' third workshop, our approach to its design and implementation, and the challenges related to framing code as material for algorithmic botanical art in the context of humanities secondary education.

2 Related Work – STEAM, Arts and Science

Theoretically, we build on STEAM, an interdisciplinary pedagogical framework leveraging art to support pupils from primary or secondary education to build interest and motivation to learn STEM subjects, science and mathematics [4]. According to Boy [5], the quest of STEAM is mostly a motivational one, in relation to the fact that most young people do not spontaneously express an interest for STEM subjects. However, STEM subjects provide competences that are much needed in our contemporary innovation-driven society and can provide young people with great opportunities for remunerative careers. Moreover, we find that STEAM studies often approach art-making as a recreational relief more than a learning resource, enabling school pupils to enjoy their science lectures [6].

Unlike typical STEAM studies, we aim at going beyond motivational goals, digging deeper in the interconnections between arts and sciences, framing art as part of the scientific method of natural sciences and CT. As according to Anderson-Tempini [1] drawing can be seen as a method to “interrogate” forms and structures of natural organisms. More specifically botanical drawing is a practice for botanical inquiry, allowing naturalists to observe closely the morphology of different types of plants, record similarities and anomalies across multiple specimens, document the development process and document in a compelling way [1, 7]. In our view, arts and sciences merge into one another in the moment artists/naturalists are driven by a fascination with the beauty of nature to pursue knowledge of the natural world, while at the same time seeking to create aesthetically beautiful representations of nature itself.

3 Third Workshop – Constraints and Design

The three Taxonomies workshops were ideated in collaboration with at a local Danish high school, Rosborg Gymnasium, which focuses on curricula in the humanities. Our contact person in the school was the art teacher, who expressed the wish of hosting a module that could combine art, natural sciences, sustainability, and the Anthropocene. He was open to investigating new hands-on lecture formats, building on art as a driving force to support learning. Our creative process started from the quest of tying art, natural and social science together, building on our respective knowledge and research interests. Building around the making of natural materials and the notion of engaging the students in art-making, botanical art emerged as an ideal core practice, already embodying scientific goals and meaning. The overall project timeline, with planning and workshops’ structure is visible in figure 1.

Figure 1: Timeline of planning and structure of workshops

The third workshop goals were: to expose humanities and art students to coding, framed as an aesthetic activity, and demystifying coding as a valid tool in an art drawing context, for art-skilled students. We also had to work within a set of constraints:

- the code should be very simple, we cannot adopt "no-code" solutions, since we want to introduce CT or at least some form of algorithmic thinking,
- the code should be easy to change for humanities students, to support a hands-on format and learning-by-doing that are typical in Danish education
- minimal programming language learning required, i.e. no complex syntax nor advanced data-structures
- and the overall workshop should fit in a single slot of two times 45 minutes, a hard constraint from the school

Given that the students and teachers in art classes tend to have no programming experience, they usually would not consider coding as a natural medium for botanical art. However, in our experience, digital art and game assets are created using algorithmic-based tools: in particular Lindenmayer Systems [2] are very common in 3D graphics software and generative art tools. Blender, Maya, 3ds Max and Houdini, all have variations of L-Systems for fractals, terrain or plants generation. Therefore, we decided to look closer at L-Systems and design our workshop around them. Introduced in 1968 by Aristid Lindenmayer (a botanist), they are string-rewriting systems, very similar to context-free grammars [8], and:

"[…] were conceived as a mathematical theory of plant development. […] The emphasis was on plant topology, that is, the neighborhood relations between cells or larger plant modules." [2]

The main difference being that derivation in L-Systems is parallel, i.e. productions are not applied sequentially as in grammars, but instead production rules replace simultaneously all symbols in a given word. L-Systems have become very famous in the 1990s and have been used to draw plants, algae and grass. Deterministic and probabilistic versions of L-Systems exist, but here we focus on the former for simplicity. A classic example of L-System is shown in figure 2: the symbols are F and X, the starting symbol. The parallel derivation produces the following strings:

X => F[-X][+X] =>
FF[-F[-X][+X]][+ F[-X][+X]] => ...

According to [2] these strings can be drawn if they are interpreted as LOGO language commands [9]: '–' and '+' representing rotations of a fixed angle, F moving the LOGO turtle forward, and the brackets controlling recursion, effectively pushing and popping the state of the LOGO turtle. X can be interpreted as F but the branch has a special color, i.e. X is a sprout (as visible in figure 2).

These rewriting systems are very expressive, they can even generate iterated fractals (as in [10] chapter 8) and the resulting images are considered aesthetically pleasing. However, L-Systems do not satisfy all our constraints: we found their formal nature problematic, since their syntax is based on context-free grammars which belong in Computer Science, with formal methods and language theory; moreover, the parallel rewriting process for L-Systems is a recursive algorithm, and both parallelism and recursion are advanced topics in programming.

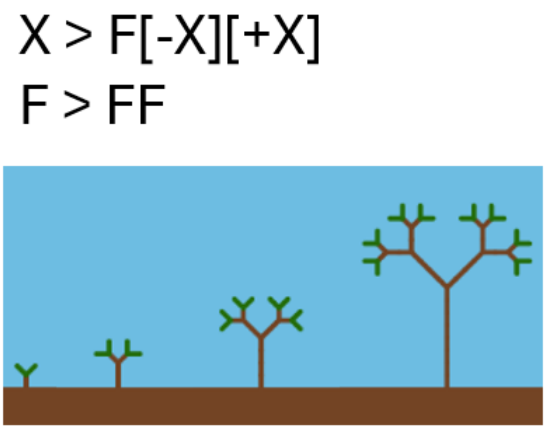

Figure 2: Derivation of a tree from L-Systems rules (at the top)

3.1 L-System Concretization Method

Humanities high-school students don't know about formal systems and would have problems understanding and using recursive algorithms. However, two of the authors have been working with tangible models for computation ([11] and [12]), so we decided to leverage tangibles, and represented our L-Systems' elements (branches, leaves and possibly flowers) as cards. This makes the rewriting and the rendering simpler to explain and easier to simulate manually: the students only have to place cards on the table, match and change some of them, following the rules.

We designed a general method to convert any L-System (or context free grammar in fact) to a set of cards with textual and/or visual rules, that can be manually executed to simulate the rewriting system algorithm; we call it L-System concretization method. We devised two versions of this method: in both we take the L-System and convert it into a deck of cards and a set of rules, but in the first version the rules are purely textual, while in the second version the rules are allowed to be more visual and based on placing and relative angles of the tangibles. In version one, given a L-System $\mathbf{G = (V \cup C, s, P)}$ where:

- $\mathbf{V}$ is the alphabet
- $\mathbf{C} = \{ -, +, [,] \}$ are the control symbols
- $\mathbf{s}$ is the start symbol
- $\mathbf{P}$ is a set of production rules of the form $\mathbf{a \text{->} B}$ with $\mathbf{a \in V}$ and $\mathbf{B \in (V \cup C)^*}$

A deck of cards is created, with a card for every symbol in $\mathbf{V \cup C}$, then every production rule $\mathbf{a \text{->} B}$ is shown by using a concatenation of cards (figure 3). Derivation is done by replacing a card with one or more cards, according to the left and right hand of the production rules; and rendering requires to simulate the behavior of a LOGO turtle, using the control symbols to rotate, move and push/pop the state of the turtle.

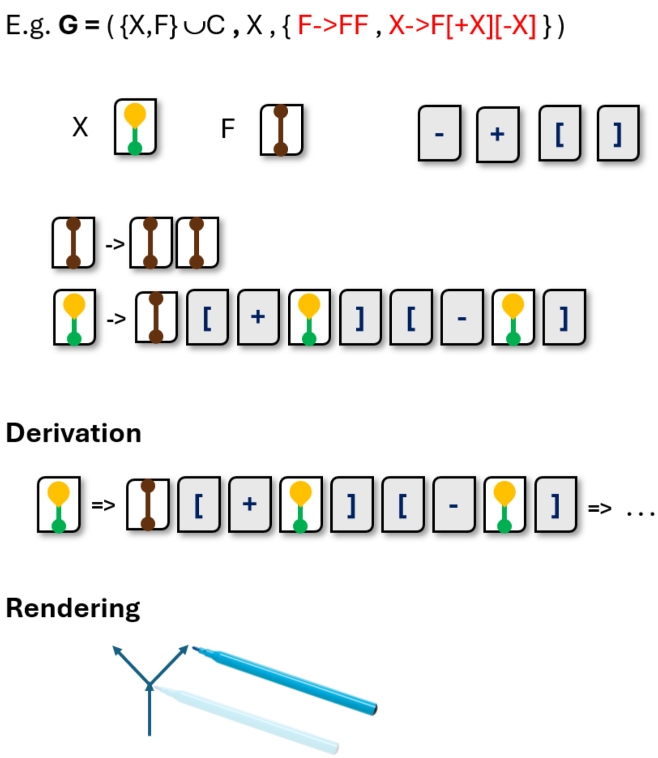

Figure 3: Example of concretization, version one of our method

In version two, the L-System G will be concretized in the same way as with version one, but the control symbols are absent, and the production rules are represented visually, by relative positioning and angles between cards (see figure 4). The advantage is that students can directly lay their plants out, without reasoning in terms of LOGO commands.

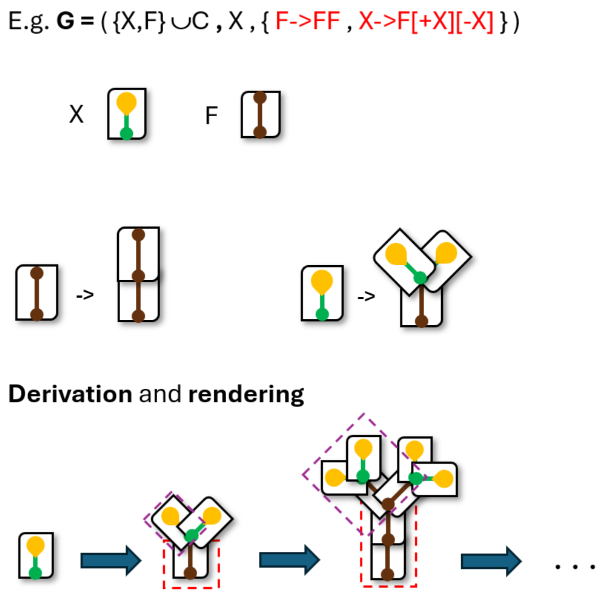

Figure 4: Concretization using the second, the “visual version” of the method

We used both versions of the method for our workshop, starting from the second (see how the students use the visual rules in figure 5 on the right). The first method required us to explain how the LOGO forward and rotate commands work, as well as when to lift the drawing pen (i.e. the LOGO turtle) and when to push and pop the state of the pen. We did not use the terms push and pop with the students, and instead explained that they have to remember where the last branching occurred in the plant they were drawing, and back-track to that: so instead of trying to explain the stack data structure to them, we used the parentheses tangibles to control the LOGO drawing pen. The results of this manual algorithm simulation are visible in figure 5, on the left.

3.2 L-System Simulator

Apart from tangibles, we also wanted to provide the workshop participants with an online app to explore L-Systems. Therefore, we implemented an L-System simulator in P5.js (freely available at https://editor.p5js.org/andrea270872/full/3Z0R3PxZW) and made it available to the students. In this software simulator a student can change the L-System parameters, but also write new sets of rules and see them in action. This is not coding, since it does not involve understanding and using the actual syntax and semantics of a universal programming language, but still requires an understanding of a recursive rewriting algorithm and the implications of how changing rules reflect on the procedurally generate plants.

We also implemented a simpler version of a specific L-System in Scratch (freely available at https://scratch.mit.edu/projects/969810191/), in case any of the students knew that language and were interested in understanding more of the coding of L-Systems. Very few of the students remembered having used Scratch in elementary school, and nobody was interested in looking at the visual code. For this implementation we used the LOGO extension and unfortunately, we had to use cloning to implement some basic form of recursion.

Figure 5: Students working with tangibles (on the right), and drawing plants following LOGO commands (on the left)

4 Running the Third Workshop and Findings

During all the three workshops, data was gathered through ethnographic observations, note taking, sketching and photos of the students' experiments with colors, drawing, and the material on L-Systems [13]. The gathered data was subject to a qualitative, interpretive analysis [14], as we identified themes emerging from the data. Finally, interviews with the teacher and the students, provided us with feedback about their experience, including suggestions for running new workshops.

A total of 28 students participated in the workshops (in April 2024), all female between 18 and 20 years old; 20 students were present at the 3rd workshop. During the first part of the workshop the students received a light lecture about L-Systems, supported by slides, showing examples of visual rules and derivation of plants. Our observations show that the students seemed interested, but toward the end of this theory part, they appeared slightly intimidated, bored, and started to get distracted. After the lecture, we moved to the second part, in which the students were introduced to the first task: to cut the cards needed for a specific L-System, and perform a few steps of derivation, creating a growing plan with the tangibles and the visual rules (as visible in figure 5 on the left). In general, the students started cutting the cards quickly and managed to assemble the code within 5-10 minutes in groups. The students then were given a choice between using the prepared cards or drawing the plants with pencil and paper, using the LOGO-based algorithm; most students used the cards, but several also drew by hand. One used the computer, drawing her plant in a graphic program instead of paper. The LOGO algorithm was found more challenging by the students, they asked questions and needed more support, hence they took longer to get started. They seem a bit confused by the commands (rotations and angles in particular), but eventually most students found the correct answer, i.e. drew their plant correctly.

In the third part of the workshop, the students got to play with and explore the online simulator (see figure 5 on the right). They got very excited, quickly started changing parameters, generating all types of plant variants: some looked realistic, while others were funny and surrealist (see figure 6). The students were engaged with the simulator, they laughed, showed each other what they made, and discussed. One student created very naturalistic plants and seemed to have a good understanding of the code, i.e. the meaning and effect of the rules.

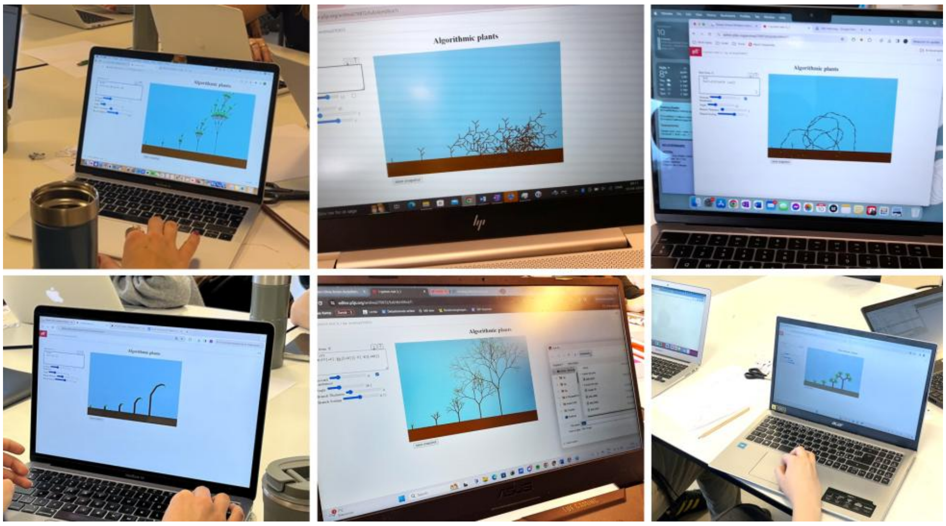

Figure 6: Examples of plants created during the workshop

Overall, we observed the class starting slowly, but the engagement of the students increased quickly as the workshop progressed. It was interesting to notice that the students appeared a bit intimidated during the initial lecture, but displayed a livelier engagement during the second and third parts, as these included mainly active exercises. Interestingly as the students engaged with the paper material, they chatted with each other and played with the cards in small groups, on the other end the simulator on the computer afforded a more individual engagement, leading the students to focus as the students explored various forms of plants through modifying the given parameters and even the code.

4.1 Discussion

During and after the 3rd workshop, informal interviews were conducted asking the students about their experience. The interviews were initiated with a simple question such as: “How was it for you to work with L-Systems?”. In general the students answered saying: “It was fun and really cool to see how it works, the outcome”, “the program was fun”, “it was very different than what we normally do” and “it was a way more structured way of thinking and it was nice to put art and plants that are usually organic into a structure, we are not used to it in art class”. A few comments pointed out the fact that in the 3rd workshop the images of the plants were created from a model, while in the previous workshops art skills were applied to draw images from reality. In this sense, the outcome from the 2nd and 3rd workshops were found substantially different in terms of visual quality and meaning of the gained representation of plants; a sample of the drawings created in the 2nd workshop are shown in figure 7. Moreover, historically, botanical art had a representational goal, summarizing features of real plants, and going from particular to general, concrete to abstract. In this sense, we could argue that botanical painting as a scientific method is inductive at its core, as naturalists were generalizing from the depiction of an individual specimen towards its species. On the other hand, L-Systems can be seen as deductive, as individual images of plants are generated from an abstract archetype representing the generalization of a species, more than a specimen with its own specificity.

Figure 7: Botanical drawings created by the students in the second workshop

During our dialogue with the students, we asked if the students could see coherence among the three workshops of the Taxonomies project and all students nodded in agreement, a few pointed

out that they could see that, although different, L-Systems could be another art material to represent plant forms, as the workshops were shifting from analogue and hand-crafted to digital and computer generated images.
Finally, the students suggested improvements to the L-System simulator, in particular they were interested in generating animations instead of static pictures of their plants, so that they could see how specific plants might grow and develop. A valuable outcome from the workshop, was that L-Systems provided the students with a basis to tinker and gain an understanding of simple visual grammars, while acquiring some basic understanding of algorithmic thinking.

5 Conclusion

The third workshop in the Taxonomies series was challenging to design and implement. This paper discussed how the resulting materials and structure of the workshop supported our goals: in only one and a half hours, we showcased creative and aesthetic coding to a class of students with no previous coding experience. The engagement grew during the lecture, and was eventually strong towards the end; the students comments and our observation of their behavior during the workshop showed a positive engagement with our simulator and the code, which we interpret as a sign that our prototype contributed to a demystification of coding and elicited creative aesthetic explorations of the potential of L-Systems as a valid tool in a botanical art drawing context.

Our workshop structure leverages tangibles, manual algorithm animation, and a light form of coding with our L-System simulator. Our workshop format, theme and simulators (paper and digital) are suitable for humanities students, and can well support open STEAM discussions with them. The teacher also provided positive feedback, and we are looking into running new iterations of the 3 workshop series, which will include the involvement of more teachers and subjects; for instance, the chemistry teacher volunteered to participate and cover the chemistry of making natural colors as part of the new edition of our first workshop.

Future work aims at the generalization of all 3 workshops into a kit that other teachers can run by themselves in other high schools. Specifically, for the 3rd workshop the kit will also include a teacher-level explanation of our L-System concretization method, this will take into consideration that many high-school teachers who would use the kit will probably not be CT experts. We are also looking at presenting L-Systems as aesthetic coding in Danish primary school; for this we are designing a 3D version of the L-System concretization method, that could be used for primary schools with modeling clay, and could help pupils create simple 3D plant sculptures, in connection with CT and 3D printing topics that are currently being introduced in Danish primary curricula.

Acknowledgement

We wish to thank Jonas Wolter, teacher, and Hanne Hautop rector at Rosborg Gymnasium, as well as Peter Dahl leader of Fablab at Spinderihallerne.

References

[1] G. Anderson-Tempini, Drawing as a Way of Knowing in Art and Science, Intellect Books, 2017.

[2] P. Prusinkiewicz, and A. Lindenmayer, The algorithmic beauty of plants, Springer Science & Business Media, 2012.

[3] G. W. Flake, The computational beauty of nature: Computer explorations of fractals, chaos, complex systems, and adaptation, MIT press, 2000.

[4] I. K. Namukasa, J. Hughes, & R. Scucuglia, "STEAM and critical making in teacher education," In Handbook of cognitive mathematics, 2022, pp. 939-970. Cham: Springer International Publishing.

[5] G. A. Boy, "From STEM to STEAM: toward a human-centred education, creativity & learning thinking," Proc. of the 31st European conference on cognitive ergonomics, 2013, pp. 1-7.

[6] S. Mejias, et al., "The trouble with STEAM and why we use it anyway," Science Education, Vol. 105(2), 2021, pp. 209-231.

[7] I. Neacşu, "On Contemporary Botanical Art," Bulletin of the Transilvania University of Braşov, Series IV: Philology & Cultural Studies. 2023;15(Suppl): pp. 25-34.

[8] N. Chomski, "On certain formal properties of grammars," Information and control, vol. 2, issue 2, 1959, pp. 137-167.

[9] C. Solomon, et al., "History of logo," Proc. of the ACM on Programming Languages 4 HOPL 2020, pp. 1-66.

[10] D. Shiffman, The Nature of Code: Simulating Natural Systems with JavaScript, No Starch Press, 2024, pp. 427-435.

[11] A. Valente and E. Marchetti, "Programming Turing Machines as a game for technology sense-making," Proc. 2011 IEEE 11th International Conference on Advanced Learning Technologies, IEEE, pp. 428-430.

[12] A. Valente, and E. Marchetti, "Playful Learning and shared Computational Thinking: the PaCoMa case study," Proc. International Conference on Computers in Education - ICCE 2020, Vol. II, pp. 180-189. Asia-Pacific Society for Computers in Education.

[13] D. Kharel, "Visual ethnography, thick description and cultural representation," Dhaulagiri Journal of Sociology and Anthropology, vol. 9, 2015, pp. 147-160.

[14] P. Dourish, Reading and interpreting ethnography, Ways of Knowing in HCI, 2014, pp. 1-23. New York, NY: Springer New York.