Using Cultural Viewpoint Metaphors in the Analysis of Computational Thinking Teaching

Gabriela A. Araujo de Oliveira
Federal University of Minas Gerais.
Computer Science Department
Belo Horizonte, Minas Gerais
gabriela-amaral@ufmg.br

Oto Braz Assunção
Federal University of Minas Gerais.
Computer Science Department
Belo Horizonte, Minas Gerais
oto.braz@dcc.ufmg.br

Raquel Oliveira Prates
Federal University of Minas Gerais.
Computer Science Department
Belo Horizonte, Minas Gerais
rprates@dcc.ufmg.br

ABSTRACT

There are several experience reports on teaching computational thinking (CT) to children in a playful way. However, we did not find in the literature any studies presenting a well defined proposal of types of tools or methods that could be adopted to teach different concepts of computational thinking. Therefore, we investigate whether Cultural Viewpoint Metaphors (CVM) could be a good way to analyze, classify, and indicate which CT tools and teaching approaches existing studies are addressing. We first analyze existing research on teaching computational thinking to middle school students in Brazil. Then, the tools and methods used in those works in light of CVM. Our findings show positive results regarding the possibility of using CVM as a foundation to classify tools and methods for teaching CT.

CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**;

KEYWORDS

Cultural Viewpoint Metaphors, Computational Thinking

ACM Reference Format:
Gabriela A. Araujo de Oliveira, Oto Braz Assunção, and Raquel Oliveira Prates. 2019. Using Cultural Viewpoint Metaphors in the Analysis of Computational Thinking Teaching. In *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems (IHC 2019), October 22–25, 2019, Vitoria - ES, Brazil.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3357155.3358466

IHC '19, October 22–25, 2019, Vitoria - ES, Brazil

ACM ISBN 978-1-4503-6971-8/19/10…$15.00
https://doi.org/10.1145/3357155.3358466

1 INTRODUCTION

According to the Brazilian Computer Society (in Portuguese, *Sociedade Brasileira de Computação* - SBC) [48], in contemporary society, having basic knowledge about computing is as important for life as having basic knowledge of math, philosophy, and other sciences. Several countries such as the United States, Finland, England, Norway, South Korea, and Australia already include the teaching of computing in K-12 education [26].

In 2006, Wing presented to the community her opinion on the need of teaching *Computational Thinking* (CT) to children as it enables the development of essential skills for people living in the 21st century [62]. As stated by her, these skills include technical skills, logical reasoning, analyzing and solving real world problems, critical thinking, and creativity. The term *Computational Thinking* was coined and defined by Wing as the process of modeling and solving problems systematically and efficiently using concepts that are fundamental to computer science [62, 63].

Although there are several tools for teaching CT to children, mainly tools for teaching programming in a playful way [25], we did not find in the literature any studies presenting a well-defined proposal to classify or differentiate the diverse existing resources and teaching approaches being used to teach CT to children. Thus, in this context, our goal is to investigate whether the Cultural Viewpoint Metaphors (CVM) [42] could be useful to analyze and classify resources and teaching approaches concerning CT teaching to middle school[1] students in Brazil.

CVM are an epistemic tool of Semiotic Engineering[2][18] that enables a gradual approximation of users' native culture to a culture unknown to them. Previous research works have considered computational thinking as a new culture being taught to children and explored CVM as a tool to evaluate CT teaching systems or to plan a workshop with teachers related to a mediating system for CT teaching [30].

[1]In Brazil, classified as Fundamental II level.
[2]Semiotic Engineering [18] is an HCI theory that considers interaction as a designer to user communication mediated by computer. In this context, an epistemic tool is defined as one that helps designers' reflect about the problem's space and potential solutions designers are considering.

In order to conduct our investigation on the use of CVM, our first step consisted in identifying resources and teaching approaches being used in the Brazilian context to teach CT to middle school students. To do so, we carried out an analysis of several research works about CT and programming education in Brazilian schools. The studies were selected from a mapping study [46], which analyzed CT publications from 2001 to 2016, and complemented with an updated set of studies we collected, which were published from 2017 to April 2019. For each study, we identified which materials (computer systems or other resources), and teaching approaches were used. Then, we analyzed each one of them and classified them according to one of the CVM.

Our results show that using CVM to analyze and classify CT teaching materials and approaches brings practical and theoretical contribution to both Human-Computer Interaction (HCI) and Computer Education communities. The usage of CVM as an analytical tool contributes to the HCI community as it helps to consolidate CVM's application in this respect. The results of the analysis also help designers to understand and plan for the CT teaching systems, as well as encourage them to use CVM as an epistemic tool. To the computer teaching community, it shows the potential CVM have as an overarching classification for CT teaching materials and approaches. The analysis of different materials and approaches can help those interested in teaching CT to choose the most suitable tools and approaches for their context.

The remainder of this work is organized as follows. Section 2 presents CVM. Next, in Section 3, we present the related work regarding Computational Thinking in Brazil, and how CVM has been used in other contexts as an analytical tool. Section 4 describes the methodology used in this study followed by our analysis results described in Section 5. In Section 6. we present a discussion of our results. Finally, Section 7 presents our conclusions and future work.

2 CULTURAL VIEWPOINT METAPHORS

Cultural Viewpoint Metaphors (CVM) is a Semiotic Engineering epistemic tool for supporting Human-Computer Interaction (HCI) designers to reflect about how to present cultural aspects to users through systems interfaces [44]. CVM lead designers to reflect on how they would like to allow for users to approximate to the new culture through the system being designed.

In light of the CVM, a culture is considered a trip and users are considered travelers [42]. CVM propose a gradual approximation of users' native culture to a culture unknown by them, considering that users move from a scenario where they are isolated in their native culture to one where they are completely immersed in a foreign culture. CVM consist

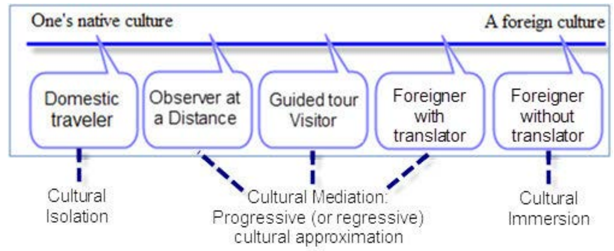

Figure 1: Progression of Cultural Viewpoint Metaphors [43]

of five metaphors as shown in Figure 1. Each metaphor represents a different level of approximation to a new culture. The metaphors are based on two main criteria: 1) interface language (native or foreign) and 2) interface signs that occurs during the interaction [43]. Table 1 shows the metaphors and their cultural variables.

	Cultural Variables	
Metaphor Expression	Language	Cultural Practice
Domestic Traveler	User's	User's
Observer at Distance	User's	User's
Guided Tour Visitor	User's	Foreign
Foreigner with Translator	User's	Foreign
Foreigner without Translator	Foreign	Foreign

Table 1: Metaphors' cultural variables

The first metaphor is called **domestic traveler**. In this metaphor, users are completely immersed in their native culture. They are in contact with language and signs of their original cultural practices.

The next metaphor is the **observer at distance**, which indicates that users are still immersed in their native culture, but the foreign culture is slightly presented to them as information. Hence, users' native language and signs of their original culture are used; however, they are contrasted with signs of the foreign culture.

Next, the **guided tour visitor** metaphor defines that users are in slight contact with a foreign culture as it is being "illustrated" to them with elements of the native culture. At this point, users start comparing both cultures. This metaphor includes foreign signs and cultural practices, but users' native language.

The following metaphor is called **foreigner with translator**. It represents users are immersed in a foreign culture, but there is still a translation into their native language. The signs and cultural practices are of the foreign culture, but the language is translated into the user's native language.

Finally, the fifth metaphor is the **foreigner without translator**, in which users are completely immersed in a foreign culture as if they were natives, without being in contact with their native culture. The language as well as the signs are exclusively of the foreign culture.

In the context of this research, CT is the foreign culture being introduced to children. Thus, cultural isolation is when students do not have any knowledge of CT, and culture immersion is when they can act within this culture without any mediation. Each metaphor represents stages in moving towards being able to interact in the CT culture without any mediation. Taking this approach, we investigate whether CVM could be useful to describe and classify CT teaching approaches and tools that have been used in the literature according to how much immersion they offer to or require from students.

3 RELATED WORK

In this section, we present the related work organized according to two different topics: Brazilian K-12 Education and how CVM has been applied. In the first subsection we describe other works that have presented an overall analyses of existing computational thinking studies in Brazil. In the second one, we report on other studies that have used CVM as an analytical tool.

3.1 Computational Thinking in Brazilian K-12 Education

Santos et al. [46] conducted a very complete mapping study of CT and programming in Brazilian K-12 education. They collected studies from 2001 to 2016 and classified a total of 338 selected works according to year, venue, study type (e.g., experience report, case study), educational stage (e.g., middle school, high school), methodological contexts (e.g., traditional, unplugged, robotics), and used tools and programming languages. The results showed that the number of publications since 2015 has been increasing, which means the interest on this research field has risen since then. This was also observed by Blatt et al. [4]. Santos et al. also observed experience reports are the most common study type being conducted in Brazil. Among 338 studies, 193 were experience reports. Consequently, there are more research topics, study types, and gaps to be addressed.

Furthermore, as in the United States, there is a vast literature of studies using Scratch to teach CT in Brazil, focused mainly on programming. Some studies show that Scratch is the tool that is used the most in Brazilian basic education [4, 31, 54, 61]. Based on this scenario, Eloy et al. [21] mapped publications that address the teaching of CT with Scratch. They collected studies published in Portuguese language between 2012 and 2016. A total of 53 papers were assessed according to teaching-learning goals, context and target audiences, strategies of evaluation, and institutions. The results showed that 49 different Brazilian institutions have conducted research with Scratch for teaching programming or CT. Besides, 31 studies were applied in middle school, and 18 in high school. Questionnaires were the most common strategy to evaluate the experiences.

Another relevant systematic review of the literature was performed by Silva et al. [55]. This work aimed to understand teaching-learning strategies of CT in Brazil. A total of 62 papers from 2012 to 2017 were classified according to their venue, Brazilian regions, teaching-learning approach (computer science unplugged, makers activities, digital electronics, distance education, programming tools, visual programming tools, digital games, educational robotics), educational stage, CT skills (abstraction, algorithms, data analysis, automation, data collection, decomposition of problems, parallelization, simulation, data representation). The results showed that the most common teaching strategy is the use of visual programming tools, followed by unplugged activities. Also, abstraction and decomposition of problems are the most explored skills respectively.

3.2 Usage of Cultural Viewpoint Metaphors

CVM and the Semiotic Inspection Method (SIM) were used together to carry out a research on PAR, an electronic game to instigate the collaboration between children with Autism Spectrum Disorder (ASD) [52]. In the light of the difficulties children with ASD have to collaborate with others, "collaboration" was considered as foreign culture unknown by children with ASD. The authors used SIM to inspect PAR, and based on CVM, they redefined collaboration patterns that were present in the game.

In another context, Leitão et al. [28], analyzed the communication of cultural aspects of interactive systems related to death. They inspected Super Lachaise, an app for visiting the cemetery Père-Lachaise in Paris. The five CVM were used as categories to guide the inspection, aiding in the identification of the main strategies designers of the app used to expose users to a foreign culture.

Mota's [30] PhD thesis focused on Polifacets, a social technical environment to support the teaching of CT. This environment has a set of facets to express AgentSheets' concepts. AgentSheets [37] is an environment that uses block-based visual language to teach programming. It is targeted primarily at children. The author used CVM to organize workshops with teachers and students to evaluate and improve communicative strategies employed on Polifacets interface.

In our previous work, we used the CVM along with the Semiotic Inspection Method (SIM) to analyze and compare the software Scratch and AgentSheets [33]. Scratch [38] is another environment for teaching programming. Similar to AgentSheets, Scratch is also targeted at children and uses block-based visual language to make it easier for them to learn how to code. CVM were used to characterize the strategies each system uses to introduce programming to their users. Our results showed that both Scratch and AgentSheets do not introduce programming in a progressive way. However, they use specific strategies in different parts of the system, and those strategies could be associated with one of the CVM. Besides, there were indications that programming can be considered as a new culture being presented to children.

The papers presented in this section show that there are several systematic studies that analyzes reports on CT education according to different aspects (study type, CT tools, CT skills, education level). However, to the best of our knowledge, there are no work that deeply analyzes a large set of materials and approaches being employed for teaching CT in Brazil. Also, CVM are a versatile epistemic tool. Although they were created to support designers, there are also research works, which inspired our research, that applied them in the context of education.

4 METHODOLOGY

The methodology used in this research is divided into four steps, which are illustrated in Figure 2, and detailed next.

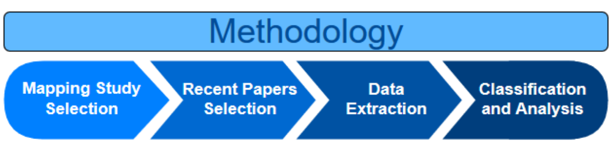

Figure 2: Methodology

4.1 Mapping Study Selection

We first searched for existing mapping studies of computational thinking in Brazil. We found 30 studies that conducted systematic literature reviews or mapping studies of computational thinking in Brazilian K-12 education. Some of them aimed to understand the general context of CT in Brazil [4, 5, 31, 46] and others presented a more specific focus, such as robotics [12], games, and unplugged activities [41]. Then, we used the following criteria to choose which of the mapping studies was better for our research:

(1) **well-defined procedures:** the mapping study should have a detailed methodology, specifying and justifying decisions;
(2) **recency:** it should be a study carried out in the longest and newest period of time after 2000;
(3) **data availability:** it should make available the papers analyzed in their study;
(4) **education level specification:** it should indicate the education level the papers focused on.

The selected mapping study [46] aimed to characterize the literature on CT and programming in K-12 education in Brazil. It used the steps suggested by Petersen et al. [35], and analyzed a total of 338 primary studies published between 2001 and 2016. Among these studies, 140 referred to studies in fundamental education (elementary school and middle school). As our goal was to analyze how CT has been taught in middle school in Brazil and which materials have been used, we only considered these as candidates for our analysis. From these papers, we eliminated those that did not explain the teaching approaches and tools that were used. At the end, we identified a total of 99 studies of our interest.

4.2 Recent Papers Selection

Although the selected mapping study [46] was published in 2018, it only considered papers published until 2016. Thus, we used adapted procedures from this paper to search for primary studies published between 2017 and 2019 and select recent and relevant studies that met this research's specific goals. We adapted the original search string to consider only the education level we were interested in – middle school. As in the selected mapping study, we only looked for studies using a Portuguese search string.

The English version of the string we used is the following:

("computation thinking" OR "computational reasoning") AND ("primary education" OR "secondary education" OR "basic education" OR "middle school" OR "childhood education")[3].

The search was conducted in April 2019. We searched on Google Scholar, which returned 597 studies. Among them, we eliminated studies that fell into one of the following exclusion criteria:

(1) duplicated studies;
(2) publications focused on applying CT in education levels other than middle school;
(3) publications that were 2 pages or less;
(4) studies that did not explain which teaching approaches and tools were used.

After filtering the 597 studies, 114 studies remained.

4.3 Data Extraction

For the data extraction step, we defined 4 aspects that we considered relevant to our research: grade level, materials used, teaching approach, and activities. We considered that these aspects would be relevant for us to achieve our goal of understanding whether the use of CVM would be a good way

[3]Original string in Portuguese: ("pensamento computacional" OR "raciocínio computacional") AND ("educação primária" OR "educação secundária" OR "educação básica" OR "ensino fundamental" OR "educação infantil")

to analyze what type of teaching approaches and materials the existing studies about teaching CT in middle school are addressing. Next, we describe each aspect we addressed:

(1) **Grade levels:** all grades of middle school. In Brazil, they are 6th, 7th, 8th, and 9th grade;
(2) **Material:** computer systems and resources that were used to teach.
(3) **Teaching Approach:** the context and/or way in which the tools were used.
(4) **Activities:** the activities carried out using the materials and how they were conducted in each study.

The first author (Computer Science graduate student in the field of HCI) read all the selected papers diagonally to extract data regarding the 4 aspects we defined above, as well as register the reference information such as title, authors, and year of publication.

4.4 Classification and Analysis

The next step was to classify materials and teaching approaches used in the selected papers in light of the CVM. Besides the materials, we decided to also analyze teaching approaches because we noticed that the same tools could be used at different complexity levels. Hence, the metaphors associated with them could also be different. In order to carry out the classification, we considered as native the culture the one in which the person does not know CT and programming, whereas the foreign is the culture in which the person is versed in CT and programming.

The classification and analysis were carried out by the two first authors who are pursuing a Master's degree in computer science in the area of HCI. Only one of them had previously worked with CVM, the same researcher who selected the papers and extracted the data. However, both had studied CVM. First, each evaluator read and classified the materials and teaching approaches individually. Then, the results were triangulated to reduce possible biases and errors. The researchers discussed their classifications and resolved disagreements and doubts regarding them. A meeting with the senior researcher (third author) guiding this research was set up to resolve cases where an agreement could not be reached. The senior researcher acted as a third evaluator and had the final call on deciding which metaphors were more appropriate.

As this process took place, 14 papers from the mapping study that had been initially selected were excluded (85 remained) and 34 papers from the filtered recent papers returned by our own search were excluded (80 papers remained). The excluded papers fell into at least one of the exclusion criteria defined in the data extraction subsection. Thus, at the end of this stage, a total of 165 were analyzed in detail.

5 RESULTS

In this section, we first present an overview of materials and teaching approaches used in the selected studies. Then, we present our analysis of them in light of CVM.

5.1 Overview

The first part of our analysis was carried out to have an overall idea of the materials that have been used to teach CT. Figure 3 shows the frequency with which each material was used throughout the years. The most commonly used materials were Scratch (72), Lego Mindstorms (24), Arduino (24), and Code.org (22). A considerable number of studies (51) applied unplugged activities, part of them were from the book Computer Science Unplugged (11) and the others (40) from other resources, such as the website "Racha Cuca". Among them, there were activities researchers themselves created, unplugged games, and questions from the Brazilian Olympiad in Informatics. In addition, 20 other studies used specific materials, which were used only once among all studies we evaluated, or tools that were not computer games. We grouped those materials under the category "Other". Some examples are *Kahoot!*[4], Microsoft Office Excel, and Microsoft Office PowerPoint. Specific computer games were grouped under the category "Other games" (e.g. *Computino* [13], *SoccerCraft* [29], *The Foos* [6]).

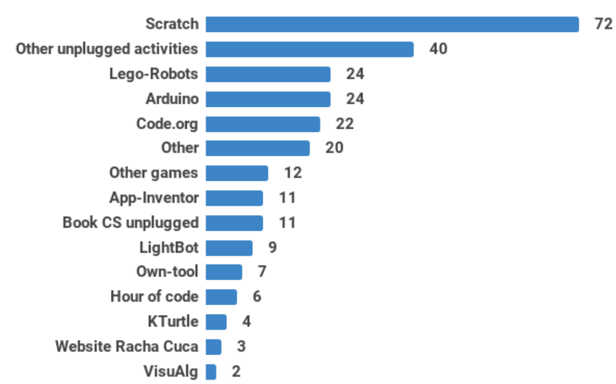

Figure 3: Usage of Materials

Figure 4 shows the application of teaching approaches before 2016 (mapping study's papers), after 2016 (papers returned by our search), and in total.

Games (74) was the most common teaching approach in the research works. In addition, a large number of studies applied **unplugged activities** (49), **animated stories** (43), and **robotics** (40). The number of studies that used games, unplugged activities, and animated stories as a teaching approach increased from 2017 to 2019. On the other hand, the number of studies that used robotics decreased in the same

[4]Kahoot!. Available at: https://kahoot.com

period of time. We can also see that a few studies utilized **traditional** (9) teaching approaches. We consider as traditional, any method of teaching in which teachers present the content mostly through lectures in classes. However, the studies that applied traditional methods did not rely solely on this approach. All of them also combined it with at least one different teaching approach. Moreover, we found 22 research works that used **other** teaching approaches. Studies whose approaches did not fit any of the aforementioned methods were grouped into the *other* category. This category was necessary because those methodologies were distinct and could not be grouped into a single category (e.g. [34, 45]).

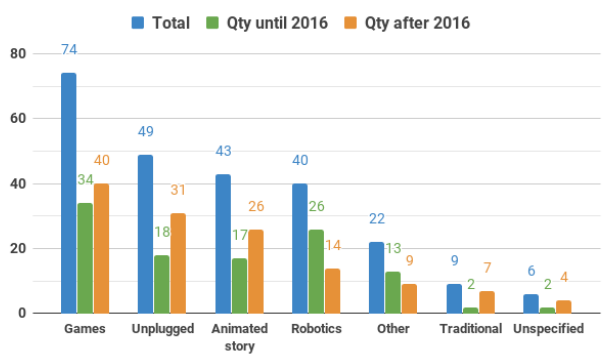

Figure 4: Teaching Approaches

Furthermore, we found a common association between teaching approaches and tools, which is depicted in Table 2. With the exceptions of **traditional** and **unplugged**, Scratch was used with every teaching approach. Also, the most commonly reported tools for teaching robotics are Lego Mindstorms, Arduino, and Scratch. Robocode and mBot are also robot related tools, but each of them were only used once.

5.2 Analysis of Materials in Light of Cultural Viewpoint Metaphors

As explained in Section 2, in the context of this research, computational thinking is a foreign culture being introduced to children. In general, we could associate unplugged activities that teach programming logic to the "domestic traveler" metaphor. However, this is true only if the unplugged activity does not require any previous CT knowledge. In other words, board games and similar other activities are part of kids' native culture, and as such are often used as a first step to introduce CT concepts to children. As they use the children's native culture (board games and language) they are generally classified as "domestic traveler" metaphor. However, if the game requires very specific CT knowledge they also fit into other metaphors. In contrast, development environments, which require users to know how to code, compile, and debug, would be associated with the "foreigner without a translator" metaphor.

In order to understand whether the CVM help with the organization of the existing CT tools, we classified the materials in accordance with the knowledge level that is explored and required for them to be used. We associated one metaphor to each of the main materials that were used in the studies we analyzed. For the analysis, we considered the materials' language, interface signs, and the context in which they were applied. We next present the most frequently used materials and how they were classified according to CVM.

Scratch[5] was associated with the "foreigner with translator" metaphor. The main reasons are that the programming language commands are translated into the user's language and are animation-oriented. For example, the command "move 10 steps forward". In addition, users can experience the programming culture by interacting with the tool, but they can still relate it to their culture. For example, the algorithms are built by snapping blocks together, which is similar to a jigsaw puzzle pieces children are used to assemble.

Similar to Scratch, **App Inventor**[6] is a block-based programming tool. It is a visual and playful programming environment that makes it easier for children to develop fully functional apps for smartphones and tablets through the use of interlocking programming blocks. So the same reasoning was applied to App Inventor, which is also associated with the "foreigner with translator".

Lego Mindstorms[7] can be associated with "foreigner with translator" or "foreigner without translator" metaphors. The metaphors can be different depending on the programming tool that is used in combination with Lego Mindstorms to program the robots. There are three options to program a Lego robot. The first option is the programming tool included in the Lego App, which is also block-based like Scratch. Also, its programming language can be translated into the user's native language. The second is using Scratch to program the Lego robots. Both of these options were associated with the "foreigner with translator" metaphor for the same reasons we previously explained for Scratch. The third option is through textual programming language. In this case, users would be totally immersed in the foreign culture because they are programming without any type of language and visual translation. Thus, this last option is associated with the "foreigner without translator" metaphor.

The same reasoning was considered to classify **Arduino**'s CVM. Similarly to Lego Mindstorms, you can program an Arduino using Scratch or using a programming language like *C*.

[5]Scratch. Available at: https://scratch.mit.edu/
[6]App Inventor. Available at: https://appinventor.mit.edu
[7]Lego Mindstorms. Available at: https://www.lego.com/en-us/mindstorms

Teaching Approaches Versus Tools

Teaching Approaches	*Tools*			
Games	Scratch	Code.org	App Inventor	LightBot
Unplugged	CS Unplugged	Racha Cuca	Unplugged games	Other Activities
Robotics	Scratch	Lego Mindstorms	Arduino	Robocode
Animated Story	Scratch	App Inventor	-	-
Other	Scratch	K-turtle	VisuAlg	-
Traditional	Other	-	-	-
Unspecified	Scratch	K-turtle	-	-

Table 2: Teaching Approaches Versus Tools Used in Middle School

Hence, Arduino can be associated with both “foreigner with translator” and “foreigner without translator” metaphors. In other words, for physical artifacts like robots and Arduino, the programming tool is the main factor we considered to classify them.

Code.org is a platform that offers a variety of courses teachers can choose according to their students' level and age. Each course has different lessons about specific topics, and each lesson is made of a set of levels that students must complete. All the studies who used Code.org explored the same level of difficulty. Most studies used the prebuilt courses (primarily course 2[8]) offered in the platform. Activities in course 2 let the users experience programming in a friendly manner as they are required to use a limited set of blocks of commands to build algorithms to solve small problems. Although these activities require children to use coding blocks as does Scratch, only a small amount of commands is available to solve an instantiated problem (e.g. define the commands for a character to move from one specific position to another in a board). Thus, we associated these types of activities to the “guided tour visitor” metaphor because they presented the new culture (programming commands), but guided their use in a limited scope. In these activities, students did not have the option to build their own project or use their creativity to define different tasks for the characters.

A considerable number of studies used the book **Computer Science Unplugged**[9]. The book is composed of 12 activities that are structured in three different levels of knowledge, which are basic, intermediate, and advanced. The basic level addresses activities about data. The intermediate level covers different topics involving algorithms. Finally, the last level presents more advanced subjects: finite automata and programming languages. The research works we evaluated only applied activities of the first and second level, such as “Count the Dots", which is about binary numbers, and “You Can Say That Again”, which is about text compression. Although these activities are unplugged and part of the easier levels, they present very specific computer science subjects to students. Hence, these type of activities and involved knowledge were associated with the “guided tour visitor” metaphor because children can experience the foreign culture through unplugged materials, but they cannot practice them freely (running and debugging a program in a digital platform).

LightBot[10] is a paid app for teaching programming logic through a puzzle game. However, there is a LightBot: Hour of Code version available for free, which is what was used in the research works we analyzed. The goal of the game is to move a robot around the scenario to light up all the blue squares. Three levels (basics, procedures, loops) and five commands (move forward, turn left, turn right, jump, light up) are available on the Hour of Code version. LightBot is a simple game, and its commands are presented in a visual way. Children would not think they are practicing programming logic while playing it unless they are told so. Besides, it is translated into multiple languages, including Brazilian Portuguese. Hence, we associated the “guided tour” metaphor to LightBot.

Table 3 presents a consolidated analysis of tools and their metaphors. Regarding the metaphors, the analyzed materials vary from “guided tour visitor” to “foreigner without translator”. Among the most used materials analyzed, none were classified as “domestic traveler” or “observer at a distance”. In our analysis, we found a few unplugged activities classified in these two metaphors (e.g. [2, 19]). These resources were often used as an introduction to some of the CT basic concepts, before moving on to other activities associated with the other three metaphors.

As expected, most tools adopted a “guided tour visitor” to “foreigner with translator” approach as these metaphors indicate that children are exposed to and explore some CT concepts without requiring them to have a full grasp of the whole

[8]Code.org. Available at: https://studio.code.org/s/course2

[9]CS Unplugged. Available at: https://csunplugged.org/en/

[10]LightBot. Available at: http://lightbot.com/

Tools Versus Cultural Metaphors

Tools	*Cultural Metaphors*	
Scratch	Foreigner with Translator	-
Arduino	Foreigner with Translator	Foreigner without Translator
Lego Mindstorms	Foreigner with Translator	Foreigner without Translator
Code.org	Guided Tour Visitor	-
App Inventor	Foreigner with Translator	-
CS Unplugged	Guided Tour Visitor	-
LightBot	Guided Tour Visitor	-

Table 3: Tools Versus Cultural Metaphors

computing culture. Furthermore, these systems normally utilize playful elements of children's native culture, such as games and animations. Finally, from the most frequently reported tools, only Arduino and Lego Mindstorms were classified as "foreigner without translator", which would require the student to be fully immersed in CT culture. However, depending on the programming tool being used to program them, they could also be classified as "foreigner with translator". In fact, the "foreigner with translator" approach was more frequently used with them, which could be a scaffold for children before they can be totally immersed in the required programming languages.

5.3 Analysis of Teaching Approaches in Light of Cultural Viewpoint Metaphors

While we analyzed the tools previously presented, we noticed that the level of cultural approximation explored by each study could vary in accordance with the teaching approach being applied. For instance, although a tool was classified as "foreigner with translator", teachers could prepare a very structured activity and it ended up promoting a "guided tour visitor" approach. Hence, we also used CVM to evaluate the teaching approaches being applied in the studies.

Among the studies that used **Scratch**, most of them (46) first give an overview of the tool to introduce it and then conduct practical activities. We identified two types of introductions: one where the teacher gives an oral explanation (e.g., lecture) about the tool, and another where children were given guided tasks to perform using the system. For the first approach, we considered the "observer at distance" metaphor because the foreign culture is communicated to the children without them experiencing the culture themselves. On the other hand, the second approach is associated with the "guided tour visitor" metaphor because the children could experience the tool, but through a structured activity. Other studies using Scratch (26) let the children explore the tool by themselves without any previous introduction. In this case, the activity's classification was the same of Scratch's, as it did not constrain the students in any way.

Most studies (11) that used **Arduino**, first utilized a more intuitive and playful tool before using Arduino. Among those tools, authors reported using Scratch, Lego Mindstorms, and Code.org. In this case, the teaching approach's metaphors are the same of the tools. For example, studies that first used Scratch and then Arduino have associated with them the "foreigner with translator" and Arduino's metaphors ("foreigner without translator" or "foreigner with translator"), respectively. In other studies, children went straight to the activities to develop scripts for Arduino using visual block language (6), or using textual programming language-based tools (7). These activities were classified as "foreigner with translator", and "foreigner without translator", respectively.

Studies that used **Lego Mindstorms** normally did not combine it with another tool. Most of them (13) used Scratch or a programming tool included in the Lego App to program the robots. In these cases, the metaphor associated with the approach is the same as the Lego's metaphor, "foreigner with translator". Some of the research works (7) first give a lecture explaining and guiding children about how to use the tools before introducing them to the actual activities with Lego Mindstorms. For those, the teaching approach's metaphors were "guided tour visitor" and Lego Mindstorms' metaphor, which can be "foreigner with translator" or "foreigner without translator" depending on the programming tool. In a few studies (3), children went straight to the activities to build Lego robots using textual programming language-based tools.

Research works on CT often start with teaching CT through the use of unplugged activities, and then move on to digital tools. [15, 20, 40, 47]. In the studies we analyzed, the most common tool used after unplugged activities is **Code.org**. Unplugged activities that stimulate the development of creative and logical thinking without a direct relation to computer science concepts are associated with the "domestic traveler" metaphor. For example, activities available at the website "Racha Cuca"[11]. On the other hand, unplugged activities that address computer science concepts, such as logical programming, are associated with the "observer at distance" metaphor. Hence, studies that used Code.org have the "domestic traveler", "observer at distance", and "guided tour visitor"(Code.org's metaphor) associated with them.

6 DISCUSSION

In this section, we discuss the results obtained from the tools and teaching approaches' analyses based on CVM.

CVM represent a continuum of cultural approximation plotted with reference to a presumed user's native culture. It goes from a cultural isolation ("domestic traveler") to a complete cultural immersion ("foreigner without translator"). In light of CVM, none of the systems by themselves fully present an introduction to computational thinking (the foreign culture) to the users in a gradual manner. Gradual approximation was usually achieved by using more than one resource, or by creating activities that gradually moved from a limited usage of a system that required more knowledge (e.g. classified as "foreigner with translator"), to using all of its resources (and its metaphor in full).

In systems like Scratch, App Inventor, Scratch for Arduino, and Blockly, the programming commands are presented as jigsaw pieces which intends to facilitate the user's understanding of block commands that can be put together or not. Moreover, the programming commands are presented in natural language, which helps the users to understand them. Although the "foreigner with translator" strategy adopted in these systems facilitates children's grasp of some aspects of CT related to programming, they still require students to have achieved some knowledge of the programming culture to use it creatively. Thus, in the light of CVM, educators who plan to fully support students' gradual learning should use other simpler resources before moving to more complex systems such as Scratch, App Inventor and Arduino.

There were many studies where computational thinking was introduced to children gradually [7, 8, 11, 14, 17, 22, 23, 40, 56, 59, 60]. Most of them proposed first the application of unplugged activities and then the use of digital tools such as Code.org, LightBot, Scratch, and other specific games. The following methodology is commonly employed by researchers: 1) application of unplugged activities from the book Computer Science Unplugged and other sources; 2) use of existing digital games such as LightBot and Code.org lessons which does not require children to create scripts from scratch; 3) creation of new games or animated stories using mainly Scratch and other visual language-based tools. In general, teachers adopted a cultural approximation that initiated at a "domestic traveler" and/or "observer at distance" approach based on unplugged activities, transitioned to tools that explored a "guided tour visitor" approach, and finally used tools associated to the "foreigner with translator" metaphor strategy. There were a few studies that went all the way to a "foreigner without translator" strategy. Grasping programming logic and CT concepts by itself is hard even for undergraduate students, which may lead them to drop computer science and other similar courses [24, 32]. Hence, a gradual teaching approach is especially important when the students being taught are young.

Studies that present a concern with the introduction of computational thinking gradually when using a specific tool were conducted as well [1, 39, 49, 51]. They presented activities with increasing difficult levels. For example, in a study with Arduino [36], the activity began with a simple LED activation, followed by making two LEDs blink alternately, and finishing with the development of a traffic light system. In this study, children were also supported by a guide with the steps to be followed. Similarly, there were also studies that adopted this strategy for teaching children how to use Scratch, using activities to introduce programming concepts progressively [16, 27, 50]. They first explained concepts such as variables, then introduced control and repetition structures, and finally proposed the creation of a game or animation. These approaches show concern regarding presenting concepts gradually to students. However, the complexity of tools such as Arduino and Scratch might be an obstacle for students. They do not have elements of the initial CVM. Hence, using and understanding those systems may be harder for students who are younger or have yet to experience and get used to programming logic.

Regarding the complete cultural immersion, which corresponds to the "foreigner without translator" metaphor, only studies with Arduino and Lego Mindstorms took a step in that direction. Users are fully immersed in cases where textual programming languages are used to program robots and scripts. There is no translation into the user's native language and cultural practices. However, among all studies with Arduino and Lego Mindstorms, only a few of them adopted this strategy. Furthermore, some research works did not immerse students straight into an environment without translator; rather, they presented a visual block language first [3, 9, 10]. In these studies, authors introduced coding in a "foreigner with translator" approach, and supported students in passing on to the "foreigner without translator" metaphor little by little. From our corpus of 165 papers, only 3 (namely, [53, 57, 58]) reported children being directly immersed in "foreigner without translator" without mentioning any strategies regarding a gradual introduction of some level.

In addition, the results presented in 5 indicate there is not a preferable approach to teach computational thinking

[11]Racha Cuca. Available at: https://rachacuca.com.br

concepts to middle school students in Brazil. Games (electronic or not) are the teaching method researchers used the most. Unplugged activities are also widely explored. Considering Brazil's current reality, the high amount of studies that use unplugged activities is not surprising. Many Brazilian public schools still lack resources, especially technological ones, for teaching non-classical subjects [41]. Therefore, research works that employ unplugged activities are valuable since they can promote computational thinking education to everyone.

On the other hand, almost half of the studies we analyzed used Scratch. Scratch is free to use and can be used without internet connection if it is locally installed. Besides, it is a flexible tool that makes it possible for activities that take a more gradual approach in terms of cultural approximation to be carried out before using Scratch to its full potential. Studies with Scratch in Brazilian schools often report positive results [21], which indicates that Scratch is a suitable tool and should be considered by educators who intend to teach programming or CT in Brazilian schools that are equipped with computer laboratories.

7 CONCLUSIONS AND FUTURE WORK

This paper presented a study to investigate whether CVM are capable of supporting the analysis and classification of tools and approaches used in computational thinking education. We selected a set of materials (computer systems and other resources) and teaching approaches being used to teach computational thinking in Brazilian middle schools, through a systematic analysis of the literature. Each one was classified as one of CVM metaphors based on the cultural approximation they provided, considering the continuum between the students native culture (not having learned any CT concepts) to the foreign culture (in which they are fluent in CT). Our classification using CVM allowed us to describe relevant aspects of the materials analyzed, as well as teaching activities, regarding their cultural approximation of CT and distinguish between them. This classification alone is interesting as it informs teachers (or other stakeholders interested in CT material) what level of CT knowledge (cultural approximation) is expected from children to use a given material.

Furthermore, our analysis indicated that for some materials, teachers used activities that led them to gradually expose students to the material's full cultural approximation. In other words, teachers created structured or guided activities that helped introduce children to the material. Other studies reported strategies in which different materials were used to gradually move students to a higher level of immersion in CT culture.

Thus, we can conclude that CVM is a useful analytic tool and classification scheme for CT material and teaching approaches, as the result of our analysis can be relevant to researchers, education professionals, and system developers. In terms of research, using CVM allows for a conceptual classification of the existing resources and approaches that does not depend on the devices adopted or even CT concepts explored. Hence, it can be useful in comparing or proposing strategies to approach, or means to compare different CT concepts. Although in this work we focused on CT for middle school, the education level of the material bears no influence on the CVM classification.

For education professionals, the results of the classification can be used to support them in selecting the materials and teaching approaches that would be appropriate for their students' level and educational purposes, as well as defining their own strategies for introducing CT to their students. Also, it could be useful if for any reason an ideal material could not be used in their context, to identify others that are classified in the same metaphor and would be suitable candidates to substitute that ideal one.

Finally, for system developers, the existing classification could be helpful for them to understand the different systems aimed at CT teaching, identifying niches in which there is an opportunity for them to contribute with a new system. Furthermore, it could help them better understand the use of CVM in the CT context, and encourage them to use it as an epistemic tool when developing their system.

This investigation is of an interdisciplinary nature, as it explores solutions created for the HCI field to analyze education resources regarding CT teaching. Thus, it contributes to the consolidation of CVM as an analytical tool. Although CVM was proposed as a design epistemic tool, it is interesting and useful to have indicators of how it can be broadly used. It also contributes to the field of computer education, specifically aimed at CT teaching – theoretically, as it shows the relevance of the use of CVM as an analytical tool and classification scheme, but also with practical results generated by the analysis of different resources and teaching approaches.

The next steps in this research include exploring the use of CVM and results of the analysis to guide the proposal of a teaching methodology to introduce computational thinking to middle school children. We plan to carry out an action research to investigate whether a methodology that promotes gradual approximation to the foreign culture of computational thinking can have positive effects on learning.

ACKNOWLEDGEMENT

Oto Braz Assunção thanks the National Council for Scientific and Technological Development (CNPq) for financing his research.

REFERENCES

[1] Luemy Avila, Flavia Bernardini, and Patrick Moratori. 2016. O uso de robótica para aprendizado de programação integrando alunos de

educação básica e ensino superior. In *Anais do 24 Workshop sobre Educação em Computação*.

[2] Simone Bavaresco, José Francisco Szücs, and Cláudia Santos Fernandes. 2018. *Aplicação de uma metodologia para o ensino de ciência da computação para crianças e para estudantes do curso de bacharelado em ciência da computação.* Technical Report.

[3] Fábio de Lima Bezerra and Klissiomara Dias. 2012. Programação de computadores no ensino fundamental: experiências com logo e scratch em escola pública. (2012).

[4] Lucas Blatt, Valdecir Becker, and Alexandre Ferreira. 2017. Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação.. In *Anais do Workshop de Informática na Escola*. 815.

[5] Adriana Bordini, Christiano Avila, Monica Marques, Luciana Foss, and Simone Cavalheiro. 2017. Pensamento computacional nos ensinos fundamental e médio: uma revisao sistemática. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 123.

[6] Rafael Marimon Boucinha, Christian Puhlmann Brackmann, Dante Augusto Couto Barone, and Ana Casali. 2017. Construção do Pensamento Computacional Através do Desenvolvimento de Games. *Renote - Novas Tecnologia na Educação* 15, 1 (2017).

[7] Lara Carlos, Julia Godinho, and Janaina Gomide. 2018. Um Relato de Experiência da Escola de Verão de Programação para Crianças. In *Anais do Workshop de Informática na Escola*, Vol. 24. 41.

[8] Débora da Conceição Araújo, Ariane Nunes Rodrigues, Cláudia Vangéssica de Araújo Silva, and Leonardo Silva Soares. 2015. O ensino de computação na educação básica apoiado por problemas: Práticas de licenciandos em computação. In *Anais do XXIII WEI (Workshop sobre Educação em Computação) Garanhuns*.

[9] Tatiana Souza da Luz Stroeymeyte, Herbert Gomes da Silva, and Carolina dos Santos Batista. 2014. Robótica com Arduino: Uma Proposta PBL Freiriana no Desenvolvimento de Competências e de Habilidades. *SIED: EnPED-Simpósio Internacional de Educação a Distância e Encontro de Pesquisadores em Educação a Distância* (2014).

[10] Débora Priscilla da Silva, Simone Sidnei, Ângelo Jesus, and Carlos Eduardo Paulino Silva. 2016. Aplicação de robótica na educação de forma gradual para o estímulo do pensamento computacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 1188.

[11] Mercedes Matte da Silva, Sandra Teresinha Miorelli, and Anelise Lemke Kologeski. 2018. Estimulando o Pensamento Computacional com o Projeto Logicando. *Revista Observatório* 4, 3 (2018), 206–238.

[12] Edjane Mikaelly Silva de Azevêdo, Deise Juliana Francisco, and Albino Oliveira Nunes. 2017. O Avanço das publicações sobre a robótica educacional como possível potencializadora no processo de ensino-aprendizagem: uma revisão sistemática da literatura. *Redin-Revista Educacional Interdisciplinar* 6, 1 (2017).

[13] Rozelma Soares de França, Waldir Cosmo da Silva, and Haroldo Jose Costa Amaral. 2013. Computino: um jogodestinado à aprendizagem de Números Binários para estudantes da educação básica. In *XXI Workshop sobre Educação em Computação*.

[14] Rozelma Soares de França, Victor Afonso dos Santos Ferreira, Luma Cardoso Ferro de Almeida, and Haroldo José Costa do Amaral. 2014. A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. In *Anais do XXII Workshop sobre Educação em Computação (WEI-CSBC)*.

[15] Fernanda de Melo Reis, Fábio Cristiano, Danielle Martins, and Patrícia da Rocha. 2017. Pensamento Computacional: Uma Proposta de Ensino com Estratégias Diversificadas para Crianças do Ensino Fundamental. In *Anais do Workshop de Informática na Escola*. 638.

[16] Hévellyn de Morais Rabêlo, Wênnio Kelson de Oliveira, Luan de Luna Santos, Ana Liz Souto de Araújo, and Flávia Veloso Souza. 2015. Scratch na produção de recursos interdisciplinares com disciplinas indígenas. In *XXIII Workshop sobre Educação em Informática*.

[17] Millena Lauyse Silva de Oliveira, Anderson Alves de Souza, Aline Ferreira Barbosa, and Emanoel Francisco Spósito Barreiros. 2014. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In *XXXIV Congresso da SBC-XXII Workshop de Ensino de Computação, Brasília*.

[18] Clarisse Sieckenius De Souza. 2005. *The semiotic engineering of human-computer interaction.* MIT press.

[19] Jéssica Silva de Souza and Alba Sandyra Bezerra Lopes. 2018. Estimulando o pensamento computacional e o raciocínio lógico no ensino fundamental por meio da OBI e computação desplugada. In *XXIX Simpósio Brasileiro de Informática na Educação*.

[20] Wilk Oliveira dos Santos, Fernando Carvalho Silva, Lucas Tadeu Hinterholz, Seiji Isotani, and Ig Ibert Bittencourt. [n. d.]. Computação Desplugada: Um Mapeamento Sistemático da Literatura Nacional. *RENOTE* ([n. d.]).

[21] Adelmo Antonio da Silva Eloy, Roseli de Deus Lopes, and Isabela Martins Angelo. 2017. Uso do Scratch no Brasil com objetivos educacionais: uma revisão sistemática. *RENOTE. Revista Novas Tecnologias na Educação* (2017), 1–10.

[22] Rozelma Soares de França, Waldir da Silva, and Haroldo do Amaral. 2012. Ensino de ciência da computaçao na educaçao básica: Experiências, desafios e possibilidades. In *XX Workshop sobre Educação em Computação*. 4.

[23] Ronaldo Vaz Friedrich, Daniela Scherer dos Santos, Rodrigo dos Santos Keller, Márcio Daniel Puntel, and Daniel Biasoli. 2012. Proposta metodológica para a inserçao ao ensino de lógica de programaçao com logo e lego mindstorms. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*.

[24] Maria Martins Giraffa and Michael da costa Mora. 2013. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In *Congresos CLABES*.

[25] Vitor Gomes, Renata Pontes, Carlos Camelo, Gilvonaldo Cavalcanti, and Mirko Perkusich. 2017. Ensino de programação para crianças e adolescentes: um estudo exploratório. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 490.

[26] Fredrik Heintz, Linda Mannila, and Tommy Farnqvist. 2016. A review of models for introducing computational thinking, computer science and computing in K-12 education. In *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE.

[27] Tânia Knittel, Marilene SS Garcia, Karen Iglesias, and Diego Spitaletti Trujillo. 2018. SCRATCH: uma linguagem de construção interativa de competências matemáticas. In *XVI Simpósio Brasileiro de Jogos e Entretenimento Digital*.

[28] Carla Faria Leitão, Vinícius Carvalho Pereira, and Cristiano Maciel. 2017. Exploring the Communication of Cultural Perspectives in Death-Related Interactive Systems. In *Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems*. ACM, 26.

[29] Simão Martin, Simone Cavalheiro, Renata Reiser, Luciana Foss, Ana Rita Mazzini, André Du Bois, and Clause Piana. 2018. SoccerCraft: Relato de Atividade para Ensino Aprendizagem de Habilidades do Pensamento Computacional Aplicada no Sexto Ano do Ensino Fundamental. In *26º Workshop sobre Educação em Computação*.

[30] Marcelle Pereira Mota. 2014. *PoliFacets: um modelo de design da metacomunicação de documentos ativos para apoiar o ensino e aprendizado de programação.* Ph.D. Dissertation. PUC-Rio.

[31] Carlos Nascimento, Débora Abdalla Santos, and Adolfo Tanzi. 2018. Pensamento Computacional e Interdisciplinaridade na Educação

Básica: um Mapeamento Sistemático. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 709.

[32] Wilson Neto and Aguinaldo Schuvartz. 2007. Ferramenta Computacional de Apoio ao Processo de Ensino-Aprendizagem dos Fundamentos de Programação de Computadores. In *Simpósio Brasileiro de Informática na Educação - SBIE*.

[33] Gabriela Amaral Oliveira, Rafael Sales Medina Ferreira, and Raquel Oliveira Prates. 2018. An Exploratory Analysis of Interactive Systems for Introducing Programming Based on Cultural Viewpoint Metaphors. In *Proceedings of the 17th Brazilian Symposium on Human Factors in Computing Systems*. 7:1–7:11.

[34] Neuza Oro, Ariane Pazinato, Adriano Teixeira, and Ádler Gross. 2015. Olimpíada de Programação de Computadores para Estudantes do Ensino Fundamental: A interdisciplinaridade por meio do Software Scratch. In *XXI Workshop de Informática na Escola*.

[35] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic mapping studies in software engineering.. In *Ease*, Vol. 8. 68–77.

[36] Cássio Pinheiro, Sabrina Deloise Leal, and Celson Lima. 2015. Um Cenário Viável para Motivar os Alunos do Ensino Fundamental para um Futuro Ingresso nos Cursos Superiores de Computação. 405.

[37] Alex Repenning. 1993. Agentsheets: a tool for building domain-oriented visual programming environments. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 142–143.

[38] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* (2009), 60–67.

[39] Stephanie Ribeiro and Amanda Melo. 2017. Um Método para o Desenvolvimento de Software com Crianças Utilizando o Ambiente Scratch. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 28. 1027.

[40] Guilherme Rodrigues and Laureane Sousa. 2017. O ensino do pensamento computacional como forma de inclusão tecnológica e motivação de crianças. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 28. 1784.

[41] Sebastião Rodrigues, Eduardo Aranha, and Thiago Reis Silva. 2018. Computação Desplugada no Ensino de Programação: Uma Revisão Sistemática da Literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 417.

[42] Luciana Cardoso de Castro Salgado. 2011. *Cultural viewpoint metaphors to explore and communicate cultural perspectives in cross-cultural HCI design*. Ph.D. Dissertation. Pontífica Universidade Católica do Rio de Janeiro.

[43] Luciana Cardoso De Castro Salgado, Clarisse Sieckenius De Souza, and Carla Faria Leitão. 2011. On the epistemic nature of cultural viewpoint metaphors. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction*. Brazilian Computer Society, 23–32.

[44] Luciana Cardoso de Castro Salgado, Carla Faria Leitão, and Clarisse Sieckenius de Souza. 2013. Cultural Viewpoint Metaphors. In *A Journey Through Cultures*. Springer, 43–67.

[45] Gilson Santos and José Ricardo Mafra. 2018. O Pensamento Computacional e as Tecnologias da Informação e Comunicação: como utilizar recursos computacionais no ensino da Matemática?. In *Workshops do VII Congresso Brasileiro de Informática na Educação*.

[46] Priscila Cavalcante Santos, Luis Gustavo Araujo, and Roberto Almeida Bittencourt. 2018. A Mapping Study of Computational Thinking and Programming in Brazilian K-12 Education. In *2018 IEEE Frontiers in Education Conference (FIE)*. 1–8.

[47] Arianne Sarmento, Marcos Farias, and Higor Santos. 2017. Relato de Experiência do PIBID: Promovendo o Ensino de Computação de forma interdisciplinar com Português no Ensino Fundamental. In *Anais do Workshop de Informática na Escola*. 313.

[48] Sociedade Brasileira de Computação SBC. 2017. Referenciais de Formação em Computação: Educação Básica.

[49] Pasqueline Dantas Scaico, Mychelline Souto Henrique, Felipe Oliveira Miranda Cunha, and Yugo Mangueira de Alencar. 2012. Um relato de experiências de estagiários da licenciatura em computaçao com o ensino de computaçao para crianças. *RENOTE* 10, 3 (2012).

[50] Gelson André Schneider, Eliane Mittelstad Martins de Souza, and Luciano Ferreira Gomes. 2018. Ensino e Aprendizado de Lógica Através do Scratch. In *V Seminário Nacional de Inclusão Digital*.

[51] Carmen Pamela Rosales Sedano, Igor Vasconcelos Nogueira, Sheila Vilas Boas Fraga, and Marcos Augusto Francisco Borges. 2017. Aplicaçao do MITApp Inventor como ferramenta de apoio à aprendizagem. In *Educación científica e inclusión sociodigital: actas del IX Congreso Iberoamericano de Educación Científica y del I Seminario de Inclusión Educativa y Sociodigital*. Servicio de Publicaciones, 862–871.

[52] Greis Mireya Silva, Luciana Cardoso de Castro Salgado, and Alberto Barbosa Raposo. 2013. Metáforas De Perspectivas Culturais Na (Re) Definição De Padrões De Colaboração De Um Jogo De Multitoque Para Usuáios Com Autismo. In *Proceedings of the 12th Brazilian Symposium on Human Factors in Computing Systems (IHC '13)*. Brazilian Computer Society, 112–121.

[53] Jamille Silva, Fábio Cristiano, Danielle Martins, and Wilson da Silva. 2018. Storytelling e Robótica Educacional: a construção de carros robôs com Arduino e materiais recicláveis. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. 1806.

[54] Jailson Cunha da Silva. 2017. Ensino de Programação para alunos do Ensino Básico: Um levantamento das pesquisas realizadas no Brasil. Trabalho de Conclusão de Curso.

[55] Kennedy dos Santos Silva, Nícolas Pierim Pereira, and Valguima Odakura. 2018. Mapeamento Sistemático: estratégias para o ensino-aprendizagem do Pensamento Computacional no Brasil. In *XXIII Congreso Internacional de Informática Educativa*. 319–329.

[56] Sônia Fortes da Silva, Aline Ferreira Barbosa, Anderson Alvez de Souza, Eraylson Galdino, Millena Lauyse Silva de Oliveira, Sebastião Rogério Neto, and Wilk Oliveira Santos. 2015. Relato de experiência de ensino de computaçao no ensino fundamental em estágio supervisionado da universidade de pernambuco no campus garanhuns. In *23º Workshop sobre Educação em Computação*. 1–10.

[57] Aline Sousa, Sivaldo Silva, Alberto Raiol, João Sarges, and Fábio de Lima Bezerra. 2015. O universo ludico da programação de computadores com logo no ensino fundamental. XXIII Workshop sobre Educação em Informática.

[58] Leandro Sousa, Daniel Costa, Ana Claudia Martinez, Thiago Ribeiro, Leandro Couto, and Jefferson Souza. 2016. Ensino de programaçao em robótica móvel no ensino fundamental e médio. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. 399.

[59] Mychelline Henrique Souto, Eline Macedo, Felipe Cunha, and Pasqueline Scaico. 2013. Um relato histórico da prática docente em disciplinas de estágio contado por estudantes da Licenciatura em Computação. In *Proc. Congresso da Sociedade Brasileira de Computação-XXI Workshop sobre Educação em Computação, Maceió, AL*.

[60] Givanaldo Souza, Pauleany Simões de Morais, Jeanne da Silva Barbosa Bulcão, Anna Raquel da Silva Marinho, and Danylla Medeiros. 2016. Saberes e Fazeres da Docência na Licenciatura em Informática: Relato de Experiência das Ações do PIBID. In *Anais do Workshop de Informática na Escola*, Vol. 22. 241.

[61] Claudia Werlich, Avanilde Kemczinski, and Isabela Gasparini. 2018. Pensamento Computacional no Ensino Fundamental: um mapeamento sistemático. In *XXIII Congreso Internacional de Informática Educativa*. 375–384.

[62] Jeannette Marie Wing. 2006. Computational thinking. *Commun. ACM* (2006), 33–35.

[63] Jeannette Marie Wing. 2011. Computational Thinking–What and Why? https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why. Accessed: 2018-11-03.