Strategies to Introduce Computational Thinking to Children: An Analysis based on Cultural Viewpoint Metaphors

Gabriela Amaral Araujo de Oliveira[1], **Oto Braz Assunção**[1], **Raquel Oliveira Prates**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

gabriela-amaral@ufmg.br, {oto.braz,rprates}@dcc.ufmg.br

Abstract. *Computational thinking (CT) is regarded as an essential competence for people living in the 21st century. Hence, several materials for teaching CT have been developed. Although there are studies that classify materials and approaches for teaching CT, an analysis regarding how deeply they present computing concepts to students is still lacking. In this research, we conduct such analysis using an epistemic tool called Cultural Viewpoint Metaphors (CVM). We first analyze existing research on computational thinking teaching. Then, we classify the tools and teaching methods used in those works in light of CVM. We expect our analysis can support educators to select teaching materials for the activities they will conduct and define the order in which they will be applied.*

1. Introduction

Education has always had a crucial role in preparing the next generation for the world they are going to live and work. Currently, the development of skills related to computing, especially problem solving skills, has become necessary. Including computational thinking in basic education a way to make computing a science for all. This inclusion can happen through the creation of specific classes, or if students are required to apply computational abilities in their regular classes.

[Wing 2006] was the first researcher who defined the term Computational Thinking (CT), and made it known worldwide. According to her, CT promotes the development of important skills for people who are living in the 21st century. For example, technical skills, logical reasoning, real world problem resolution, critical thinking, and creativity. The Brazilian Computer Society (SBC) made available a proposal to include computing in Brazilian k-12 education [SBC 2017]. The document presents skills and competences related to computational thinking, digital world, and digital culture which should be addressed from elementary to high school.

There are several materials for teaching computational thinking, mainly tools for teaching programming in a playful way [Santos et al. 2018]. These tools expose students to computational thinking at different levels (i.e.: matchstick puzzles VS visual programming tools). However, to the best of our knowledge, there is not an analysis of materials and approaches for teaching computational thinking regarding how deeply they present computing concepts to students. In this research, we conduct such analysis using Cultural Viewpoint Metaphors (CVM) to classify materials and teaching approaches for teaching CT. CVM are an Human-Computer Interaction epistemic tool to support the decision of how a new culture should be presented to users. In the context of this work, computational thinking is the culture being taught to children.

DOI: 10.5753/cbie.sbie.2019.547

Considering the metaphors are a gradual way to present a new culture, our analysis and classification of materials and teaching approaches can support educators and researchers at: (i) comparing teaching materials regarding how deep they present computing concepts; (ii) choosing appropriate teaching materials for the activities they will conduct; and (iii) defining an order for tools to be used for teaching in way their complexity increases gradually.

2. Computational Thinking in Brazilian k-12 Education

There are many research works that analyzed studies focused on computational thinking teaching in Brazil through a Systematic Literature Reviews (SLR) and systematic mappings. They classify papers according to different criteria such as study type, materials used for teaching, teaching approaches, publication venue, concepts being taught, and others.

[Werlich et al. 2018] conducted a systematic mapping of research works about CT in middle schools that were published in Brazil and internationally. The authors classified the studies regarding their teaching strategies, school grade, materials, and evaluation methods.

There are also systematic studies focused on specific methodologies and tools for teaching computational thinking. [Rodrigues et al. 2018] conducted an SRL of studies focused on Computer Science Unplugged. They considered the following aspects to classify the studies: school grade, motivation for using unplugged materials, concepts being addressed, methodology, evaluation methods, and advantages and/or limitations of using unplugged materials. On the other hand, [Gomes et al. 2018] conducted a mapping of computer games for teaching. The games were mapped based on game mechanics, CT abilities they promote, and genre. Besides games, educational robotics is another methodology widely used to engage students. [dos Santos et al. 2018] carried out an SLR to investigate the robotics kits that are mostly used, and how researchers and educated use them to develop CT abilities.

Our research is not as broad as the work of [Werlich et al. 2018], but also not as specific as ones of [Rodrigues et al. 2018, Gomes et al. 2018, dos Santos et al. 2018]. The focus of our research are teaching materials and approaches in order to understand how deeply computer science and programming concepts can be taught using them. We used a conceptual tool called Cultural Viewpoint Metaphors to conduct our analysis.

3. Cultural Viewpoint Metaphors

The Cultural Viewpoint Metaphors (CVM) are an epistemic tool of the Semiotic Engineering [1] for supporting Human-Computer Interaction (HCI) designers to communicate their purpose of putting users in contact with cultural diversity [Salgado et al. 2013]. In this section, we describe the CVM and how they have been applied as an analytic tool.

The CVM consist of five metaphors of a travel through a cross-cultural territory as shown in Figure 1 [Salgado et al. 2011]. Each metaphor represents a different level of approximation to a new culture and can be plotted upon a continuum. It starts from a

[1] Semiotic Engineering is a HCI theory that considers interaction as a communication from designer to user mediated by a computer.

complete isolation of users in their native culture and go to their immersion in a foreign culture. In the context of this research, the cultural isolation involves a culture without any knowledge of programming and the culture immersion covers a culture with knowledge about programming.

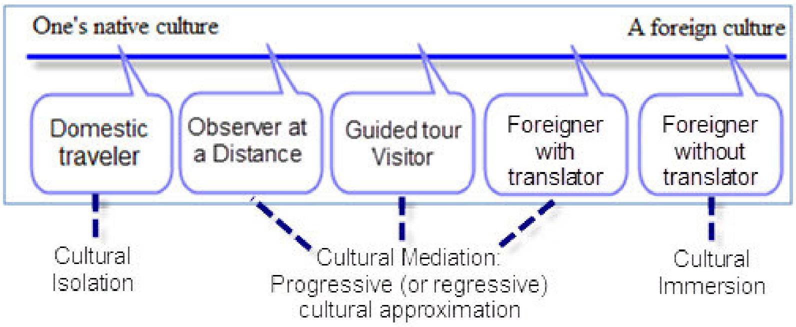

Figure 1. Progression of Cultural Viewpoint Metaphors [?]

The first metaphor is called **Domestic Traveler**. As the name implies, the designer does not aim to expose users to cultural diversity in this stage. Instead, it proposes cultural isolation, which means users are completely isolated in their native culture without being in contact with any sign of a foreign culture. In the context of CT, this metaphor is associated to logic games and unplugged activities to stimulate skills like creative thinking and logical reasoning, which are not related to any specific computer science concept. On the other end of the spectrum, the **Foreigner without Translator** metaphor intends to completely expose users to the foreign culture. Users are in a complete cultural immersion without any explanations or guides. In the context of this research, this metaphors happen when students are in contact with textual programming languages, such as C, Python, and Java.

Between these two metaphors extremes, there are the **Observer at Distance**, **Guided Tour Visitor**, and **Foreigner with Translator** metaphors. Differently from the other two, designers use these metaphors to communicate different levels of cultural approximation with the foreign culture. For the observer at distance metaphor, users are slightly exposed to the foreign culture but their culture is dominant. Designers provide cultural diversity only as information, but not experience (e.g., a teacher explaining computer concepts to students, but they do not put them into practice). Next, the guided tour visitor metaphor expresses that users are in contact with the foreign culture, but they are mediated by explanations in their own culture. For example, a guided activity on a visual programming language system where students can test the programming commands with the guidance of a teacher. Finally, the foreigner with translator metaphor indicates users are immersed in the foreign culture and the only mediation is the translation of the language into their native language. Users directly experience the foreign cultural practices. For instance, the usage of block-based visual programming language systems, where students are exposed to programming, but the commands are written in their native language).

In the context of this research, we use the CVM to classify tools and methods used in computational thinking (CT) studies. CVM has been used in other works in the context of computational thinking, but with different goals than ours. [Mota 2014] used CVM to coordinate workshops that aimed to evaluate and suggest improvements on communicative strategies employed on Polifacets interface – socio-technical system supporting computational thinking teaching. Also, we have utilized CVM with the Semiotic Inspection Method do analyze how two different systems for teaching programming (i.e., Scratch and AgentSheets) supported computational thinking education [Oliveira et al. 2018].

4. Methodology

This section presents the methodology we adopted to conduct the research. It consists of 3 steps: (i) studies selection, (ii) data extraction, and (iii) classification.

4.1. Studies Selection

We used a mapping study and our own systematic search to find relevant research work on teaching CT in Brazilian middle schools[2]. We chose to initially focus on Brazilian studies in order to further understand the national scenario and contribute to Brazilian educators.

There are many studies that map research work on CT in Brazil, we found over 30 studies in the last 3 years. We considered the following criteria to choose the study we would use in this research: (i) how recent the study is; (ii) the publication period covered in its analysis; (iii) how well the procedures conducted in the paper were explained; (iv) whether the analyzed studies could be identified and retrieved; and (v) whether it categorized the studies by their education level.

As a result, we used the work by Santos et al. [Santos et al. 2018] as the basis for our analysis. In their work, they characterized research works on computational thinking and programming in K-12 education in Brazil. They analyzed a total of 338 studies published between 2001 and 2016. As our goal is to analyze materials and approaches being used to teach Brazilian middle school students, we eliminated studies that were not focused on middle school education, remaining a total of 140 papers. Then, we eliminated studies that did not detail the materials and teaching approaches that were used. In the end, 99 relevant papers remained.

Next, we conducted our own systematic search to find relevant studies that were not covered by the mapping study. A similar methodology to the one adopted in the mapping study was used to find research works that were published between 2017 and 2019. The original search string was changed to consider only the education level we were interested in – middle school: *("computation thinking" OR "computational reasoning") AND ("primary education" OR "secondary education" OR "basic education" OR "middle school" OR "childhood education")*[3].

We looked for papers using Google Scholar in April 2019. From the 597 publications that were returned, we eliminated studies that fell into one of the following exclusion criteria: (1) duplicated study; (2) study not focused on middle school; (3) study with 2 pages or less; (4) study that does not detail teaching approaches and materials that were used. A total of 114 publications remained after filtering them.

[2] In the Brazilian Educational system, our focus was on the Fundamental II level.

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

4.2. Data Extraction

The second step was extracting relevant data from the selected studies. We defined 4 aspects we considered to be relevant for us to analyze the materials, approaches, and topics being addressed by the studies: grade level; materials; teaching approach; and activities. **Grade level** are the middle school grades (6th, 7th, 8th, and 9th grade). **Materials** are any computer tool, and resources being used to teach (i.e.: Computer Science Unplugged, Scratch, Code.org). **Teaching approaches** are the context and/or the way the materials were used (i.e.: unplugged, games, robotics). Finally, **activities** are a detailed explanation of what students did using the materials.

The data extraction was done by one junior researcher (Computer Science graduate students in the field of Human-Computer Interaction) who read diagonally the selected studies to find information regarding the 4 aspects defined above.

4.3. Classification

Finally, materials and teaching approaches were classified in light of the cultural viewpoint metaphors. We classified not only the materials, but also teaching approaches because the same tools could be used differently.

Two junior researchers classified the studies. One of them being the researcher who conducted the papers selection and data extraction processes. Both of them are pursuing a Master's Degree in Computer Science. First, each evaluator read and classified the materials and teaching approaches individually. Then, a triangulation approach was applied to reduce possible biases and errors. The evaluators discussed and resolved disagreements and doubts related to distinct classifications.

During this step, another 14 papers from mapping study were excluded (85 remained) and 34 from the 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 and Discussion

This section presents the main results from the analysis performed on the selected computational thinking studies using the CVM. We also discuss the results obtained from the tools and teaching approaches' evaluation in the CVM's view.

5.1. Analysis of Computational Thinking Tools using the Cultural Metaphors

We first gathered the materials used to teach computational thinking in order to know what is most commonly used in Brazil. There is a total of 49 different types of materials, which are listed in Table 1. The most commonly used tools were Scratch[4] (72), Code.org[5] (25), Arduino[6] (24), Lego Mindstorms[7] (24), and the Computer Science Unplugged book [Bell et al. 2002][8] (22). Many studies also used "other unplugged activities" such as the

[4] Scratch. Available at: https://scratch.mit.edu/

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

[6] Arduino. Available at: https://www.arduino.cc/

[7] Lego Mindstorms. Available at: https://www.lego.com/en-us/mindstorms

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

games “Guess Who?” and “Treasure Hunt”; steps to plant a tree, steps to organize a party, finding the shortest path, and Tangram puzzle. In addition, 10 studies specified they had traditional classes to explain theoretical concepts related to computer science or computational thinking tools. We also found games that are not popular known and are not available online, so we grouped these games in the “other game” category.

We classified the materials in light of the CVM in order to understand whether they help with the organization of the existing CT tools. The analysis was done based on the knowledge level that is explored and required for materials to be used in the selected papers. In addition, we considered the materials’ language, interface signs, and the context they were applied. The same tools could be used differently in other studies, consequently, the metaphors associated with them may vary from one to another.

Table 1 presents a consolidated analysis of materials according to the metaphors. There are a few types of materials (3) which were classified as the “observer at distance” metaphor. These resources, together with materials in the domestic traveler category (9), were often used as an introduction to some of the CT basic concepts, before moving on to other activities, usually in one of the other three metaphors. All materials that do not explicitly provide evidence of CT, our foreigner culture, were considered as being the “domestic traveler” metaphor. Some examples of them are logic problems and challenges that do not intend to explore computing, as well as games that are often known to children and do not want to promote contact with CT. On the other hand, materials that present CT to children as “information”, but not as “experience”, were classified as being the “observer at distance” metaphor. The only ones in this metaphor are other unplugged activities, classes, and activities from the Hour of Code platform.

Furthermore, “guided tour visitor” (18) was the most present metaphor, followed by “foreigner without translator” (17), and “foreigner with translator” (14). Despite the high number of materials classified as the “foreigner without translator” metaphor, most of them were present in only one study, with the exception of research works that used Arduino and Lego Mindstorms. In fact, the most frequently used metaphors were “guided tour visitor” and “foreigner with translator”. This is an expected result since those metaphors indicate children are exposed to CT concepts and practices, without being completely exposed to textual programming languages. Moreover, these tools use interactive stories, games, and animations, elements that are familiar to children.

Scratch is associated with the “foreigner with translator” metaphor since the programming language commands can be translated to children’s native languages and are animation-oriented. Although some other tools (e.g., Code.org, Blockly Game) use block-based visual programming languages as Scratch does, they include a limited set of commands children can use to solve specific problems. Hence, they were classified as being the “guided tour visitor” metaphor. Unplugged activities from the CS unplugged book were also associated with ‘guided tour visitor” as they cover specific computer science topics. In this scenario, children can experience the foreign culture through unplugged materials, but they cannot practice them freely in a platform.

Moreover, some materials were associated with two metaphors. Arduino and Lego Mindstorms can be associated with either “foreigner with translator” or “foreigner without translator”. The metaphor is defined by the programming tool being used to code

Table 1. Materials and their Metaphors

Materials (Qty)	*Cultural Metaphors*		
Scratch (72)	Foreigner with Translator	-	-
Code.org (25)	Guided Tour Visitor	-	-
Arduino (24)	Foreigner with Translator	Foreigner without Translator	-
Lego Mindstorms (24)	Foreigner with Translator	Foreigner without Translator	-
CS Unplugged (22)	Guided Tour Visitor	-	-
Other Unplugged (19)	Domestic Traveler	Observer at Distance	Guided Tour Visitor
App Inventor (11)	Foreigner with Translator	-	-
Class (10)	Domestic Traveler	Observer at Distance	Guided Tour Visitor
LightBot (9)	Guided Tour Visitor	-	-
Logic Problem (7)	Domestic Traveler	-	-
Hour of Code (6)	Domestic Traveler	Observer at Distance	Guided Tour Visitor
OBI (5)	Foreigner with Translator	Foreigner without Translator	-
K-turtle (4)	Foreigner without Translator	-	-
Programming Lang. (4)	Foreigner without Translator	-	-
Other Game (3)	Domestic Traveler	-	-
Racha Cuca (3)	Domestic Traveler	-	-
Blockly (3)	Guided Tour Visitor	-	-
Game Maker (3)	Foreigner with Translator	-	-
VisuAlg (3)	Foreigner without Translator	-	-
Monster Coding (2)	Guided Tour Visitor	-	-
RoboMind (2)	Foreigner with Translator	-	-
Code Monkey (2)	Foreigner without Translator	-	-
Construct2 (2)	Foreigner without Translator	-	-
Logo (2)	Foreigner without Translator	-	-
BloxorZ (1)	Domestic Traveler	-	-
Kahoot! (1)	Domestic Traveler	-	-
Sokodan (1)	Domestic Traveler	-	-
Computino (1)	Guided Tour Visitor	-	-
Croco's Adventure (1)	Guided Tour Visitor	-	-
EANewProg (1)	Guided Tour Visitor	-	-
Flippy Bit (1)	Guided Tour Visitor	-	-
SoccerCraft (1)	Guided Tour Visitor	-	-
Sokibin (1)	Guided Tour Visitor	-	-
CriptoLAB (1)	Guided Tour Visitor	-	-
TapRec (1)	Guided Tour Visitor	-	-
The Foos (1)	Guided Tour Visitor	-	-
NoBug's Snack Bar (1)	Guided Tour Visitor	-	-
Pinte o 7 (1)	Guided Tour Visitor	Foreigner with Translator	-
Block.ino (1)	Foreigner with Translator	-	-
Kodu Game Lab (1)	Foreigner with Translator	-	-
mBot (1)	Foreigner with Translator	-	-
RoboKit (1)	Foreigner with Translator	-	-
Pencil Code (1)	Foreigner with Translator	Foreigner without Translator	-
MineCraft (1)	Foreigner without Translator	-	-
PySchool (1)	Foreigner without Translator	-	-
Super Python (1)	Foreigner without Translator	-	-
RPG Maker (1)	Foreigner without Translator	-	-
Swift Playground (1)	Foreigner without Translator	-	-
SuperLogo (1)	Foreigner without Translator	-	-

them. Therefore, for physical materials like robots and Arduino, the programming tool is the main factor we considered to classify them. The same reasoning was used for activities from the Brazilian Informatics Olympiad. They can be solved using a block-based visual programming language or textual programming languages. Therefore, both "foreigner with translator" and "foreigner without translator" metaphors were associated with them. "Other unplugged", "other games", and "classes" were associated with multiple metaphors because they are a group of specific unplugged activities, digital games, or traditional lectures which exposed students to CT at different levels of deepness. Similarly, the platform Hour of Code has three metaphors associated with it since there is a variety of activities and materials available there. As for the *Pinte o 7* and *Pencil Code* tools, both "foreigner with translator" and "foreigner without translator" were associated with them because users can choose whether they want to code using visual or textual languages.

5.2. Analysis of Computational Thinking Teaching Methods using the Cultural Metaphors

When conducting the analyses presented in the previous subsection, we noticed that the same tool could be used in different level of approximation with the foreign culture. We also observed that there are studies where different materials are combined. Therefore, we concluded that the level of cultural approximation explored by each study could change depending on the teaching approach being applied. In this subsection, we present the teaching approaches for the most frequently used materials according to CVM.

Among the 72 studies where **Scratch** was used, in 26 of them, students started to Scratch from from the beginning. In the other 46 studies, materials to introduce Scratch and CT concepts to students are used before the children interact with Scratch. Among these 46 studies, 21 used materials associated with "guided tour visitor" (usually Code.org and LightBot), before applying Scratch. Many studies also started with teaching CT through the use of unplugged activities, mostly from the CS Unplugged book and activities developed by the authors, and then transitioned to digital tools. Thus, Scratch can be used along with materials associated with "domestic traveler", "observer at distance", and "guided tour visitor". Finally, most studies that used Scratch are examples of studies that gradually introduce CT to students.

Code.org was the second most used tool for teaching CT. Among the 25 studies with code.org, there were 8 that used nothing but Code.org. The other 17 included unplugged activities, games, and/or introductory classes before introducing Code.org to the students. Most of them also focus on teaching CT in way in which the complexity of concepts and tools increases gradually. Both Scratch and Code.org use block-based visual languages, but Code.org is limited to a specific set of commands to solve an instantiated problem. Therefore, Code.org is a good option to introduce CT to students without going too deep into computer programming. Scratch is a better choice when educators want to encourage children to be creative and learn more advanced programming structures.

On the other hand, most research works that adopted **Arduino** and **Lego Mindstorms** expose the students to a complete cultural immersion in the foreign culture. Students start to interact straight with Arduino, without using materials that are less complex, in 15 out of the 24 studies. The same happens to studies with Lego Mindstorms. The other 9 studies with Arduino and 8 with Lego Mindstorms gradually introduced CT to students through the usage of Code.org, “Hour of Code”, Scratch, logic problems, and classes. These studies may consider children to have previous knowledge about programming since educators did not give detailed explanations about Arduino, Lego Mindstorms, and programming to the students.

There are various studies that applied activities from the **CS Unplugged** book (22). Most studies (17), relied solely on the unplugged activities from the book. Only 5 studies used activities from the book combined with other materials (e.g., logic problems, LightBot), which are associated with the “domestic traveler”, “observer at distance”, and “guided tour visitor” metaphors. The main difference between unplugged activities from the book and other unplugged activities is that the book cover very specific computer science subjects, whereas the others usually explore simpler skills such as creativity and logical reasoning. Furthermore, applying unplugged activities from *CS Unplugged* before introducing Scratch to children is a common practice in the studies we analyzed.

6. Conclusion and Discussion

This paper presented an analysis of materials and approaches for teaching computational thinking in order to understand how deeply they present computing concepts to middle school students. Cultural Viewpoint Metaphors were used as an epistemic tool to support our analysis. We selected studies regarding the teaching of computational thinking to Brazilian middle school students through a systematic analysis of the literature. Then, we classified the materials they used based on the cultural approximation they provided, considering the native culture as the that does not know CT concepts, and the foreign culture as the one which is *“fluent”* in CT.

Using CVM, we were able to identify whether the foreign culture, computational thinking, is being introduced in a gradual way to children or not. In some studies, educators created guided activities that helped introduce the material to children. Other studies used different materials to gradually move students to a higher level of immersion into the CT culture. As for systems such as Scratch and Blockly, the programming commands are presented as jigsaw pieces to facilitate children’s understanding of commands that can be put together. They were often used along with simpler materials in order to support students’ gradual learning. Furthermore, we found many studies that first applied unplugged activities, and then transitioned to digital tools such as Code.org, LightBot, and Scratch.

This research has relevant results for both education professionals and researchers. First, the materials analysis results can support them at selecting appropriate materials for not only what they plan to teach, but also for their students’ level of knowledge. Second, the results of our teaching approaches analysis may be used for educators to choose how CT concepts can be introduced to students. In other words, in what order the materials should be used and what teaching approaches they can apply when using those materials.

A common methodology for teaching CT to children can be structured as follows: 1) Application of activities that are simpler and more familiar to students such as unplugged activities and logic problems; 2) Introduction of computing concepts through classes; 3) Use of digital tools where children do not have to build new scripts, such as Code.org and LightBot; 4) Application of tools with a block-based visual programming language, such as Scratch, where students are encouraged to develop projects such as games, animated stories, and learning objects.

As future work, we plan to conduct an action research with middle school students in order to develop a methodology to gradually introduce CT to children, and further analyze teaching materials. We expect this methodology will have positive impacts such as facilitating the learning process of unknown and complex computing concepts, and keeping children engaged and motivated throughout the process. Furthermore, there is an opportunity to use CVM to analyze materials and teaching approaches being employed internationally, and compare the Brazilian and other countries' scenarios.

Acknowledgement

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

References

Bell, T., Witten, I., and Fellows, M. (2002). Computer science unplugged.

dos Santos, F. E., Pereira, D. S., Godin, J. M., De Lima, J. V., Zaro, M. A., and do Canto Filho, A. B. (2018). A robótica educativa no ensino de lógica de programação: uma revisão sistemática da literatura. *RENOTE*, 16(1).

Gomes, H. A. C., Melo, J. M., Farrapo, H. P., Bonates, M. F., and Coutinho, E. F. (2018). Descrição e comparação de jogos digitais para auxiliar no ensino de programação. *Revista Sistemas e Mídias Digitais (RSMD)*, 3(1).

Mota, M. P. (2014). *PoliFacets: um modelo de design da metacomunicação de documentos ativos para apoiar o ensino e aprendizado de programação*. PhD thesis, PUC-Rio.

Oliveira, G. A., Ferreira, R. S. M., and Prates, R. O. (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*, IHC 2018, pages 7:1–7:11, New York, NY, USA. ACM.

Rodrigues, S., Aranha, E., and Silva, T. R. (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)*, volume 29, page 417.

Salgado, L. C. D. C., De Souza, C. S., and Leitão, C. F. (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*, pages 23–32. Brazilian Computer Society.

Salgado, L. C. d. C., Leitão, C. F., and de Souza, C. S. (2013). Cultural viewpoint metaphors. In *A Journey Through Cultures*, pages 43–67. Springer.

Santos, P. C., Araujo, L. G., and Bittencourt, R. A. (2018). A mapping study of computational thinking and programming in brazilian k-12 education. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–8. IEEE.

SBC, S. B. d. C. (2017). Referenciais de formação em computação: Educação básica.

Werlich, C., Kemczinski, A., and Gasparini, I. (2018). Pensamento computacional no ensino fundamental: um mapeamento sistemático. 14:375–384.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.