An Ontology based approach to teach Computational Thinking

Cristiana Araújo
Centro Algoritmi / Dep. de Informática
Universidade do Minho
Braga, Portugal
decristianaaraujo@hotmail.com

Lázaro V O Lima
Centro Algoritmi / Dep. de Informática
Universidade do Minho
Braga, Portugal
lazaro.lima@ifb.edu.br

Pedro Rangel Henriques
Centro Algoritmi / Dep. de Informática
Universidade do Minho
Braga, Portugal
prh@di.uminho.pt

***Abstract*— This paper is focused on the teaching/learning process of Computational Thinking at primary and secondary schools. It is generally accepted that Programming is a complex task that requires a long learning process. Theoretical knowledge about fundamentals on algorithms and data structures, as well as, on programming languages are required but are not enough; practicing a lot is also necessary. However, teaching Computer Programming is a hard job, most of the times unsuccessful. To overcome all the difficulties, felt by teachers and students, an increasingly bigger community of researchers in Computer Science is defending the importance of teaching Computational Thinking to young students to train them, since very earlier, in logic and abstract reasoning for problem solving. Our starting point to approach this topic relies on the use of an Ontology (OntoCnE) that describes in detail the concepts "Computational Thinking" and "Programming", and maps those concepts to different education levels, starting with the first year. We believe that a person just acquires a new way of thinking, or a new way of behaving, if he is trained with the appropriate learning resources. So a main investment to educate people in Computational Thinking is on the choice/creation of those convenient resources. In particular we intended to investigate the impact of Augmented Reality in the usefulness of the referred resources. In that direction we will also discuss the development of a Web Platform to help on collecting and classifying (according to the referred ontology) learning resources to be used by teachers in computing classes. On the other hand, the platform will help on the retrieval, from that repository, of the most adequate resources to teach a specific subject to a specific level.**

Keywords — computational thinking, programming, learning resource, teacher support tools, ontology

I. INTRODUCTION

Computer programming (CP) is complex and arduous; it requires a lot of effort, perseverance, strategy and systematic work. So it is also important to research new ways on how to teach people to program properly. However, according to Paula Tavares [1], *teaching computer programming* is a difficult and most often unsuccessful job. This happens because programming requires many different skills, such as trial and error approaches, game strategies, abstraction and logical thinking; overall, it demands strong motivation.

The difficulties of learning to program are known due to high failure in the programming courses. For many students, the difficulties begin when they have to understand and apply some abstract concepts of programming, like control structures to create algorithms to solve concrete problems. The difficulty in solving problems is the most notorious deficit in students because it requires many skills that they often do not have, namely: understanding the problem (they can not interpret); prior knowledge relationship (do not establish similarity with known problems); reflection on the problem and solution (write the answer without thinking about it); and lack of persistence (they give up the problem whose solution they can find quickly and simply). In addition, students have limited knowledge of math and logic and lack of motivation [2].

Corroborating what was said above, the causes for programming learning problems, identified by this second author, lead to the assertion that students arriving at computer programming courses do not have the basic skills, such as problem-solving, reasoning, logic, abstraction, etc., that they should have acquired in previous years.

We believe that the solution for these problems is to teach and train Computational Thinking (CT) from an early age. The so-called *Computational Thinking ability* promotes the development of logical reasoning, the relationship of knowledge, abstract thinking, critical thinking, problem solving strategies, persistence, among others. If all children acquire and train these skills from childhood, they will find it easier to overcome difficulties in programming courses. It is important to point out that these skills acquired with CT are not only for students who want to be programmers, they are transversal skills to all areas.

Therefore, in this work we will present an ontology that describes the domain of CT and whose objective is to aid in the classification of learning resources that will serve to teach/train all the skills that CT promotes.

In Section II we present Computational Thinking and the importance of training it. The ontology, OntoCnE, which describes the domain of CT is presented in Section III. In Section IV the importance of Learning Resources (LR) to teach CT is presented, and some examples of resources are illustrated. The Web Platform, Micas, that assists in the storage and classification of resources is discussed in Section V. Finally, Section VI concludes the paper and proposes directions for future work.

II. THE IMPORTANCE OF TEACHING COMPUTATIONAL THINKING

In the 1960's, a computer scientist, Alan Perlis, was one of the first to recognize that it was important for all students in all areas to learn programming and "computer theory" according to Grover discourse in [3].

978-1-7281-3182-5/19/$31.00 ©2019 IEEE

It is well known that, in 1967 Seymour Papert, created the language Logo defending the idea that computer programming aided the process of building knowledge and developing thought as can be found in [4].The possibility that computing helps the child think better was evident in his book *Mindstorms* [5], in which Papert states that Logo programming can stimulate what he called *"Powerful Ideas"* and *"Procedural Knowledge"*. For Papert, computers should be used so people could "think with" machines and "think about" their own thinking.

Later Jeannette Wing, in 2006, came up with the concept *Computational Thinking*. According to Wing, **Computational Thinking** is a method for solving problems, or designing systems and understanding human behavior, based on the fundamental concepts of computer science. Wing states that Computational Thinking has already begun to influence many courses, from Sciences to Humanities, even daily life and will be a key skill used by everyone in the 21st century [6]. Teaching Computational Thinking will bring new challenges for education, especially for the early years. There are already models to teach physics and mathematics, but there are still no models to teach Computational Thinking, but we have the computer that is a tool that allows precisely to improve the learning of that topic (Computational Thinking) [7].

Computational Thinking uses the following processes to solve a problem (see Figure 1) [8]:

- **Logical reasoning:** use the existing knowledge of a system to make reliable predictions about its future behavior;
- **Algorithms:** construct a sequence of instructions or a set of rules to do something;
- **Decomposition:** break the complex problem into smaller, simpler parts to solve;
- **Patterns:** identify similarities or characteristics between problems and solve the problem using solutions previously defined in other problems and based on previous experiences;
- **Abstraction:** identify what is important and remove unnecessary details;
- **Programming:** write a set of instructions;
- **Evaluation:** ensure that the solution, be it an algorithm, system or process, is adequate to solve the problem.

In addition to the previously mentioned processes there are also some approaches that characterize Computational Thinking and that are important to develop when one begins to think computationally. The approaches that characterize Computational Thinking are [8]:

- **Tinkering:** experimenting and playing;
- **Creating:** design and do with creativity;
- **Debugging:** finding and fixing errors;
- **Persevering:** keeping going and never giving up;
- **Collaborating:** working as a team.

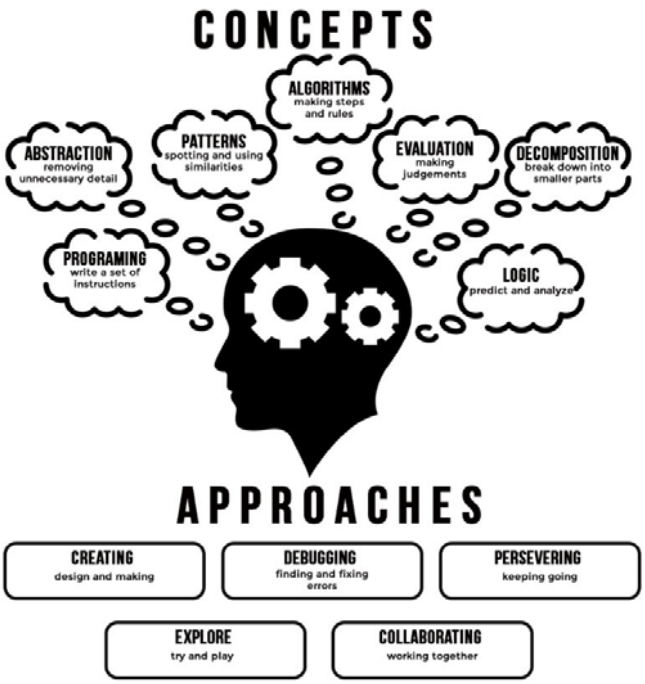

Fig. 1. Computational Thinking (adapted from: Computing at School) [8]

Computational Thinking is an important skill at all ages in the 21st century and should be developed since the childhood. The development of Computational Thinking assists on the organization of thinking and problem solving in an efficient and strategic way (it can be used in different contexts and areas); develops creativity, abstraction ability, problem decomposition, pattern recognition, critical thinking, and logical reasoning. It also helps on digital literacy.

Moreover, Computational Thinking will be crucial to capacitate people for Computer Programming. It is well known that Computer Programming is very hard to learn and teach this topic is an unsuccessful task. At present we believe that a mind trained to think computationally will learn faster and easily to program.

III. The Description of Computational Thinking using an Ontology

Computational Thinking domain is composed of several concepts that are related to each other. So we decided to create an ontology to rigourously define it in order to be possible to state precisely how to teach Computational Thinking and the stuff used to train it at various levels of teaching.

According to Gruber, *an ontology is an explicit specification of a conceptualization* [9]. Guarino describes an ontology *as an artifact of engineering, consisting of intentional vocabulary related to a certain reality, together with explicit assumptions in the form of first-order logic, representing concepts and relationships between concepts* [10].

As previously mentioned, our concern is to train Computational Thinking and for this we collect a set of

concepts that describe that knowledge domain. In Figure 2, a fragment of the built-in list of concepts[1] is displayed.

```
Conceitos{
    Abstraccao , AfericaoResultados ,
    Algoritmo , Artefacto , ArtefactoFisico ,
    Atividade , Bloco , Computador ,
    Dados , Decomposicao , DispositivoDigital ,
    Instrucao , LingGrafica , Linguagem ,
    ModoExecucao , PensamentoComputacional ,
    PensamentoEstrategico , Problema ,
    Programa , Programacao , Processo ,
    RaciocinioLogico , ReconhecimentoPadroes ,
    Resultado , Robot , Tarefa }
```

Fig. 2. Concepts that describe the domain of Computational Thinking (fragment)

After defining the concepts, we added relationships between them and construct a set of triples '(Subject, Predicate, Object)' where Subject and Object are concepts and Predicate is a relation. As a final result we obtained the ontology that describes the domain of Computational Thinking, which we designate *OntoCnE -- Ontology for Computing-at-School*[2]. A fragment of the set of triples built is presented in Figure 2. Notice that both fragments shown are written in our formal language, OntoDL [11], a DSL designed specifically to make easy the process of defining and instantiating an ontology.

```
Triplos{
Algoritmo=[
    requer=> PensamentoComputacional ,
    e_composto_por=> Bloco ,
    materializado_por=> Programa];
ArtefactoFisico =[
    isa=> Artefacto];
Bloco =[
    e_composto_por=> Instrucao];
Computador =[
    isa=> DispositivoDigital];
DispositivoDigital =[
    isa=> ArtefactoFisico];
LingGrafica =[
    isa=> Linguagem];
PensamentoComputacional=[
    requer=> AfericaoResultados ,
    requer=> Decomposicao ,
    requer=> Abstraccao ,
    requer=> ReconhecimentoPadroes ,
    requer=> RaciocinioLogico ,
    requer=> PensamentoEstrategico];
Problema =[
    requer=> Resolucao ,
    resolvido_por=> Programacao];
Processo =[
    SYN=> Tarefa ,
    e_descrito=> Algoritmo];
Programa =[
    e_composto_por=> Bloco ,
    e_realizado=> ModoExecucao ,
    actua_sobre=> Artefacto ,
    processa=> Dados ,
    produz=> Resultado ,
    e_descrito=> Linguagem];
Programacao =[
    requer=> Algoritmo];
Resolucao =[
    requer=> Processo];
Robot =[
    isa=> DispositivoDigital];
Tarefa =[
    e_composto_por=> Atividade] }
```

Fig. 3. Triples of the ontology in the Computational Thinking domain (fragment)

Reading the triples, it becomes clear how the singular concepts should be linked together to make an understandable discourse, a description that has sense. In the ontology shown in Figure 2 we can see that `Problem =solved_by=> Programming` (line 25), `Programming =requires=> Algorithm` (line 37) and `Algorithm =requires=> Computational Thinking` (line 3). Or that `Algorithm =composed_of=> Block` and `Block =composed_of=> Instruction` (lines 4 and 9). Also we learn that an `Algorithm =realized_by=> Program` (line 5) and a `Program =acts_on=> Artifact` (line 32). Moreover, we see that a `PhysicalArtifact =isa=> Artifact` (line 7) and a `DigitalDevice =isa=> PhysicalArtifact` (line 13); it is also stated that a `Robot =isa=> DigitalDevice` (line 40).

Our goal is to adapt Computational Thinking training to various levels of education. To do this, we selected the concepts to be addressed in a given level of education[3] and created relationships[4] that define the depth with which each concept shall be taught in this level of education. Then we could add to the ontology triples that describe what to teach at in each educational level to train Computational Thinking. In Figure 4, a fragment of the ontology is presented that contains the concepts that we plan to teach in the first year of the first cycle.

```
Triplos{
ano1 =[
    desenvolve=> PensamentoComputacional ,
    desenvolve=> RaciocinioLogico ,
    desenvolve=> Abstraccao ,
    apresenta=> Problema ,
    introduz=> Algoritmo ,
    introduz=> Instrucao ,
    introduz=> Programa ,
    introduz=> DispositivoDigital ,
    introduz=> LingGrafica ,
    usa=> Computador ,
    usa=> Robot ];
}
```

Fig. 4. Concepts taught in the 1st year (fragment)

In Figure 4 it is possible to see that in 1.st year (`ano1`), for example, we can introduce (`introduz`) concepts like Algorithm (`Algoritmo`), Instruction (`Instrucao`), and Graphic Language (`LingGrafica`); at this level, the teacher shall use (`usa`) a Robot, for instance; the effect in the students, expected from such teaching approach is the development (`desenvolve`) of Abstraction (`Abstraccao`) and Logical reasoning (`RaciocinioLogico`).

In the second year, some concepts that were taught in the first year shall be deepened, while new concepts shall be introduced. Again, we reinforce the idea that the level of deepening for each concept is determined by the relationship that makes up the triple. Figure 5 shows the

[1] Please notice that we will keep the ontology vocabular in Portuguese because we are building it to teach in Portugal.

[2] From the Portuguese term *Ontologia para Computação na Escola*.

[3] To deal we that idea we added to the concept list a new set of concepts `ano1` (1.st year), `ano2` (2.nd year), `ano3` (3.rd year),...

[4] Examples of such relations are: `desenvolve` (develops), `introduz` (introduce), `apresenta` (present), `usa` (uses), `reforca` (reinforces),…

concepts taught and deepened in the 2nd year of the first cycle.

```
Triplos{
ano2 =[
    desenvolve=> PensamentoComputacional,
    desenvolve=> RaciocinioLogico,
    desenvolve=> Abstraccao,
    apresenta=> Problema,
    reforca=> Algoritmo,
    reforca=> Instrucao,
    reforca=> Programa,
    reforca=> DispositivoDigital,
    reforca=> LingGrafica,
    introduz=> Decomposicao,
    introduz=> Variavel,
    introduz=> Depuracao,
    cria=> Programa,
    usa=> Computador,
    usa=> Robot,
    usa=> Teclado,
    usa=> Monitor ];
}
```

Fig. 5. Concepts taught in the 2nd year (fragment)

According to the ontology fragment in Figure 5, in the 2.nd year, the teacher will pursue the target to develop (`desenvolve`) Computational Thinking (`Pensamento Computacional`), Abstraction (`Abstraccao`) and Logical Reasoning (`RaciocinioLogico`); but he shall reinforce (`reforca`) the concept of Algorithm (`Algoritmo`), Instruction (`Instrucao`), Program (`Programa`), and Graphic Language (`LingGrafica`). In addition, the teacher will also introduce (`introduz`) Decomposition (`Decomposicao`), Variable (`Variavel`), Debug (`Depuracao`), and create (`cria`) a Program (`Programa`).

In Figure 6, a fragment of graph that depicts the complete ontology so far built, is presented.

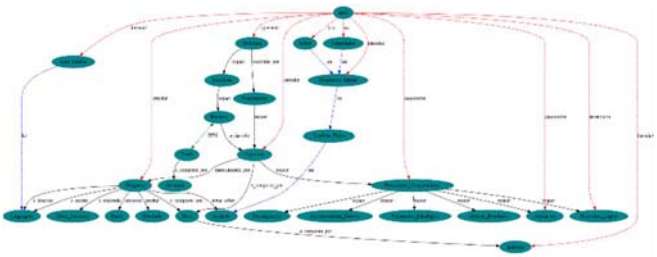

Fig. 6. OntoCnE -- Ontology for Computing-at-School

In Figure 6 the edges in red are directed towards the concepts addressed in the first scholar year. However to teach these concepts in the field of Computational Thinking it is necessary to have adequate learning resources to train the Computational Thinking.

IV. The Importance of Learning Resources

Learning Resource (LR) is *a device used for educational purposes in any format, real or virtual, that illustrates or supports one or more elements of a course of study; and may enrich the learning experience of the pupil or teacher* [12].

Learning Resources will serve to train Computational Thinking, so they play a very important role in this process. For this reason the resources must be adequate to the training of Computational Thinking so that the students can develop new knowledge and the different abilities and acquisition of values that define Computational Thinking.

To teach/train Computational Thinking we can use two types of resources: *virtual* (need electronic devices) and *unplugged* (do not need any kind of electronic devices). There are lots of well known LR yet created, but there is still space to conceive new ones more adequate to our aim. For instance an annotated text or mathematical expression markup in XML format is a simple but yet powerful artefact that can be used to train structured thinking. It introduces annotation languages in a puzzle solving style. The teacher provides the necessary "tags" for the exercise and the student has to put the tags in the correct place of the text. Each tag must have a description so that the student can interpret the tag and place it in the correct place (for example: a tag and its description `<FEITO> - is a law, a building or statue, a conquest that someone implemented`). In Figure 7 is presented a text of History, 6th year, about Lisbon Pombalina annotated with tags XML.

```
Em <DATA ano="1750">1750</DATA>, <PERSONAGEM tipo="rei">D. José I</PERSONAGEM> sobe ao trono e
<FEITO>nomeia <PERSONAGEM tipo="ministro">Sebastião José de Carvalho e Melo</PERSONAGEM>,
futuro Marquês de Pombal, como ministro</FEITO>.
<EVENTO tipo="terramoto" local="Lisboa">No dia <DATA ano="1755" mes="11" dia="1">1 de Novembro
de 1755</DATA>, <CIDADE>Lisboa</CIDADE> ficou praticamente destruída após um grande
terramoto que causou:
  - A morte de cerca de 10 000 pessoas
  - A Grande maior parte dos edifícios ficaram em ruínas
  - Perderam-se muitos tesouros como livros, manuscritos, quadros e objetos de ouro e de prata.
</EVENTO>
```

Fig. 7. Annotation in XML of a History text (6th year)

In this case the tags also have attributes, for example `<PERSONAGEM tipo="">`, which can also be used to introduce other concepts. This device can be implemented just on paper (unplugged) or on the computer (virtual).

As virtual resources we have, for example: special programming languages like *Scratch* or *Prolog*; or games like *Lightbot code hour* which is a tablet application or mobile phone that aims to command a robot to navigate in a maze and turn on the lights.

Moreover, we believe that if we add Augmented Reality to the features of our devices their effectiveness in training will be improved.

Augmented Reality was defined by Ronald Azuma in the 1990's [13] as a system that: combines virtual elements with the real environment; is interactive and has real-time processing; is designed to explore the three-dimensional space.

Augmented Reality alters the user's real world by simulating information or virtual elements in a real environment, but Augmented Reality is used to increase virtual information using all senses as well. We can then conduct explorations mixing visual, auditory, haptic, somatosensory and olfactory. An approach is considered to be Augmented Reality only if there are simultaneous interactions in the real world. The application of Augmented Reality is focused today on entertainment, but a large number of researchers has started to investigate its potential as a rich resource in education. It is curious to notice that there is a recent project [14] that extends with Augmented Reality the artefact Cody Roby (introduced above). The author integrates unplugged activities with interactions with virtual objects showing that Augmented Reality is a viable technique to train Computational Thinking.

V. CREATING A RESOURCE REPOSITORY ACCORDING TO THE ONTOLOGY -- MICAS TOOL

After recognizing the relevance of adequate learning resources to train people with the skills of Computational Thinking, it is necessary to have a support to store them in a repository duely classified. So we decided to create a Web Platform, **Micas** [15], that allows to store the resources in the repository and classifies them according to the ontology *OntoCnE* presented in Section III. In Figure 8 the system architecture is presented.

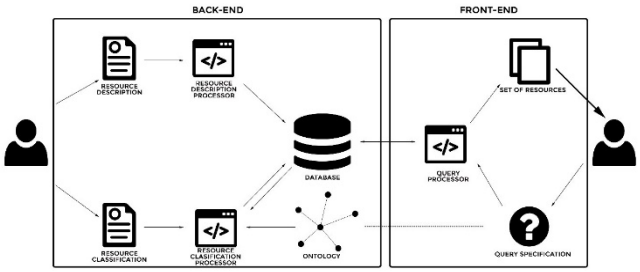

Fig. 8. System Architecture

In this web platform we will have on the one hand users who will introduce and classify new resources, and on the other hand, we will have users that will look for resources to use in their classes (see Figure 8).

OntoCnE will be used in Micas to assist the user to classify the resource being inserted. In the *Back-end*, the *Resource Description Processor* receives a *Resource Description* from the user and stores that description in the Database. At the same time, the user also makes a *Resource Classification*, which will be processed by the *Resource Classification Processor* that will be implemented based on the Ontology *OntoCnE* [15]. Below can be seen the items that constitute the description (a *fragment* for the sake of space) and the classification of the *Lightbot code hour* resource (above referred)[5]:

- **Title:** Lightbot code hour;
- **Description:** a tablet or mobile phone. . . ;
- **Type of activity:** game;
- **Type of resource:** virtual;
- **Instructions:** organize symbols/instructions. . . ;
- **Materials:** smartphone or tablet;
- **Educational level:** 2.nd year;
- **Related concepts:** ano2; Abstraccao; Instrucao; Algoritmo; LingGrafica.

The classification of the resource based on the ontology is illustrated in the field '`Related concepts`'; be aware that Micas will store, not only the concept, but also the name of the relation that links it to the specified level. The classification is very important and should be carried out with care since the chosen concepts will be the keys for any future search; those searches take into account the concept and the relation that connects it to the year specified. As already mentioned, the relation determines the level of deepening with which each concept is taught. For example if a teacher is looking for the learning resources available to introduce for the first time the concept of Algorithm (`Algoritmo`), he shall search for '`ano1 introduz Algoritmo`'.

On the *Front-end*, the *Query Processor* module is intended to receive the user specification for the type of resource he is looking for and to send the respective query to the Database. The Database, in conjunction with the Ontology, returns the results found to satisfy the specification. In turn, the *Query Processor* displays the set of Learning Resources obtained [15].

The outcome of **Micas** project is a Web Platform that allows to add new resources, list all the resources, or search for a particular set defining the scholar year, or specific keywords.

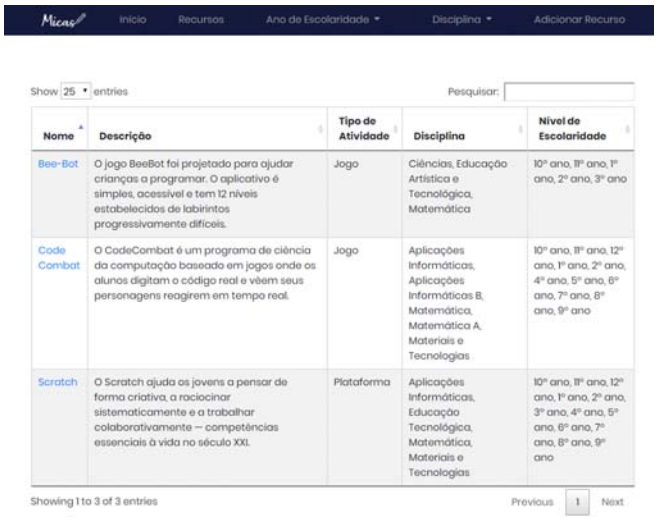

Fig. 9. Page with all the existing Learning Resources

The interface that lists all the devices available in the repository is shown in Figure 9. Figure 9 shows the page that displays the details of a Learning Resource. For more information or to use a platform see: https://micas.epl.di.uminho.pt/ .

[5] Notice that we have translated all the field names as well there content for the clarity of the paper; only the related concepts were written in pt to agree with OntoCnE vocabulary.

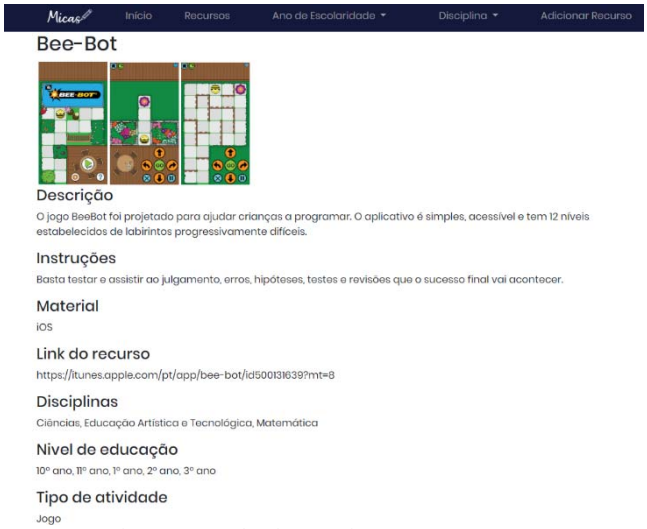

Fig. 10. Example of a Learning Resource page

VI. Conclusion

Computational Thinking is a very important concept to prepare the *21.st century Citizens*. On account of that, CT started to be taught, some years ago, in several schools of other countries; it is now time to adopt it in the primary and secondary schools of Portugal.

As it promotes the development of logical reasoning, abstraction, creativity, strategic thinking, etc., we believe that training students in that direction will help on overcoming the difficulties they face to learn Computer Programming. However, CT training should not be seen as exclusive of CP students; it brings many general skills relevant for students in all knowledge areas. So, CT must be taught as a transversal topic.

Training CT requires a smart selection of adequate Learning Resources that efficiently stimulate the acquisition of the intended skills. To assure that a convenient choice is possible, LRs must be classified based on the OntoCnE -- *Ontology for Computing-at-School*, which describes the domain of Computational Thinking. Although supported on the ontology the classification of LRs is supposed to be done manually; we do not intend at moment to build automatic tools to perform this classification. To store and classify LRs a teacher support tool, Micas, was built, as described in the paper.

Micas is a Web platform, supported on OntoCnE, that allows, on one hand, to store LRs and classify them enhancing the skills they contribute for, and, on the other hand, to search for the appropriate pedagogical devices to teach a given subject at a given level. As future work, we plan to design and conduct experiments to test Micas with primary and secondary teachers. We believe that it is important to evaluate user satisfaction with the assertiveness of retrieved resources and ease of use: teacher satisfaction questionnaires will be used for that.

Concerning the future development of Micas, we plan to implement a forum so that teachers can leave their feedback on resources and their experience with them. It is important to validate the resources and their pedagogical value prior to their usage. So Micas will include a mechanism to assure that a LR is made available only after it is validated, and not immediately after its insertion in the repository. This validation will be done by experts in the domain, highly qualified for that. In addition, we intend to continue to engage end users to discuss Micas features and to perform experimental platform testing. As explained in the paper, we are willing to contribute also to the production of modern devices that can account as LR to shape minds in CT. In that direction we intend to research the inclusion of Augmented Reality components in some devices to create new, improved, LRs as clearly motivated in Section IV. However the most important work to be carried out immediately is to enlarge as much as possible the present OntoCnE (it will never be complete) and then proceed to check its robustness.

Acknowledgment

This work has been supported by FCT -- Fundação para a Ciência e Tecnologia within the Project Scope: "UID/CEC/00319/2019". The authors want to thank our M.Sc. student Ana Azevedo for the development of Micas.

References

[1] P. C. Tavares, *O impacto da animação e da avaliação automática na motivação para o ensino da programação*. PhD thesis, Universidade do Minho, 2018.

[2] A. Gomes, C. Areias, J. Henriques, and A. J. Mendes, "Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte", *Revista Portuguesa de Pedagogia*, pp. 161–179, 2008.

[3] S. Grover and R. Pea, "Computational thinking in k–12 a review of the state of the field", Educational Researcher, vol. 42, pp. 38–43, 02 2013.

[4] S. Papert, *Teaching Children Thinking* (LOGO Memo). Cambridge, MA, USA: Massachusetts Institute of Technology, 1971.

[5] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New

York, NY, USA: Basic Books, Inc., 1980.

[6] J. M. Wing, "Computational thinking," *Commun.* ACM, vol. 49, pp. 33–35, mar 2006.

[7] J. Wing, "Computational thinking," *J. Comput. Sci. Coll*., vol. 24, pp. 6–7, jun 2009.

[8] Council for The Curriculum Examinations and Assessment, *Computing at School: Northern Ireland Curriculum Guide for Post Primary Schools*. Computing at School, 2018.

[9] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in *International Journal of Human-Computer Studies*, pp. 907–928, Kluwer Academic Publishers, 1993.

[10] N. Guarino, "Understanding, building and using ontologies: A commentary to using explicit ontologies in kbs development," *International Journal of Human and Computer Studies*, pp. 293–310, 1997.

[11] L. Martins, C. Araújo, and P. R. Henriques, "Digital Collection Creator, Visualizer and Explorer," in 8th Symposium on Languages, Applications and Technologies (SLATE 2019) (R. Rodrigues, J. Janousek, L. Ferreira, L. Coheur, F. Batista, and H. G. Oliveira, eds.), vol. 74 of *OpenAccess Series in Informatics (OASIcs)*, (Dagstuhl, Germany), p. 15:1–15:8, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

[12] R. Saskatchewan, "The Education Regulations," Technical Report, 2015.

[13] R. T. Azuma, "A survey of augmented reality," Presence: *Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[14] L. Klopfenstein, A. Fedosyeyev, and A. Bogliolo, "Bringing an unplugged coding card game to augmented reality*," INTED Proceedings*, pp. 9800–9805, 2017.

[15] A. Azevedo, C. Araújo, and P. R. Henriques, "Micas, a Web Platform to Support Teachers of Computing at School," in *Challenges 2019: Desafios da Inteligência Artificial, Artificial Intelligence Challenges* (A. J. Osório, M. J. Gomes, and A. L. Valente, eds.), (Braga, Portugal), pp. 625–641, Universidade do Minho. Centro de Competência, May 2019.