

# A Conceptual Framework for Computational Pedagogy in STEAM Education: Determinants and Perspectives

**Sarantos Psycharis\*** 

School of Pedagogical and Technological Education, GREECE  
[spsycharis@gmail.com](mailto:spsycharis@gmail.com)

**Konstantinos Kalovrektis** 

University of Thessaly, GREECE  
[kkalovr@uth.gr](mailto:kkalovr@uth.gr)

**Apostolos Xenakis** 

University of Thessaly, GREECE  
[axenakis@uth.gr](mailto:axenakis@uth.gr)

## Abstract

Computational Pedagogy is an instructional approach based on Computational Science and the Computational Experiment as well as on the CPACK model. Computational Science in Education engages students in computational modeling and simulation technology in alignment with the essential features of inquiry based teaching and learning approach and the Computational Thinking dimensions (practices and skills). STEAM –content based epistemology– education is connected to Computational Pedagogy through the Computational Experiment leading to a proposed model called ‘Computational STEAM Content Pedagogy’ as a teaching and learning approach which can be implemented in a STEAM holistic interdisciplinary/trans-disciplinary epistemology approach to the curriculum for solving real computational problems.

**Keywords:** STEM Education, Computational Pedagogy, Computational Thinking, Epistemology

♦Received 2 December 2019 ♦Revised 24 May 2020♦Accepted 24 May 2020

## Introduction

There is an increased interest for the way Computational Thinking (CT) should be implemented in the teaching and learning approach, what epistemologies should be employed, the role of simulation of models, which tools should be used, what type of data should be collected and analyzed (i.e. Guzdial, 2008). This raises also a number of questions, including how to integrate CT into the curriculum (e.g. Voogt et al., 2015; Sentence & Csizmadia, 2015; Psycharis & Kotzampasaki, 2019), how CT can be integrated with the STEM content epistemology (Psycharis, 2018), what didactic model should be proper for its inclusion, and computational methods are proper for the implementation of CT practices and CT concepts in school and Higher Education (Psycharis & Kotzampasaki, 2019). The “concept of STEM Education” needs to be clarified,

since its meaning is differentiated among researchers: “In recent years, the use of the acronym STEM (science, technology, engineering, and mathematics) has become the buzz word among the many U.S. stakeholders who have heeded the call for creating better prepared high school and college graduates to compete globally” (Breiner et al., 2012). One of the education perspectives for STEM education involves viewing the separate disciplines of science, technology, engineering, and mathematics as one unit. Even in that case, the term “unit” needs to be elaborated and answers related to the form of engagement of different epistemologies and the kind of scientific and engineering practices that should be engaged needs to be elicited. Currently, the introduction of Computational Science Education (CSE) and the computational modeling and simulation technology (CMST) –as an improvement of the technological pedagogical content knowledge (TPACK)– (Yaşar et al., 2016) pushed the inclusion in STEM content epistemology education features related to the practices like the ones employed by scientists and engineers, transforming the education settings towards the so called Computational STEM content Pedagogy (Psycharis, 2018).

## Computational Thinking (CT)

“Computational Thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science while is also considered as a universal skill and attitude that complements thinking in mathematics and engineering with a focus on designing systems that help to solve complex problems humans face” (Wing, 2006; Wing, 2008). Researchers argue that, there is no agreement for the definition of computational thinking (e.g. Barr & Stephenson, 2011; Weese et al., 2016). While computational thinking draws upon concepts that are fundamental to computing and computer science (Denning, 2007), it also includes practices such as problem representation, abstraction, pattern recognition(machine learning) decomposition, simulation, verification, and prediction (Psycharis, 2018; Psycharis & Kotzampasaki, 2019, Sengupta et al., 2013). These practices are related to the developments of models (as representations of the physical world), the scientific reasoning and argumentation as well as to Science and Engineering practices. “Despite the widespread interest in developing CT at all levels of education (and especially in compulsory education), and the increasingly large number of public and private initiatives, the successful integration of CT in school curricula is still facing open issues and challenges, such as: Is it possible to define CT as a key skill for the current century? What are its characterizing features? What are its relation to programming and computer science, on the one side, and to digital literacy, on the other? Should CT be included in compulsory education? How should skills in this field be assessed? How should teachers be prepared to best integrate it into their teaching practice?” (Bocconi et al., 2016). Breeman and Resnick (2012) introduced the so called dimensions of computational thinking, which include: computational concepts (the concepts designers engage with as they program, such as iteration, parallelism, etc.), computational practices (the practices designers develop as they engage with the concepts, such as debugging projects or remixing others’ work), and computational perspectives (the perspectives designers form about the world around them and about themselves). Moreno-León et al. (2015) redefined the “computational thinking concepts”

introduced by (Brennan and Resnick, 2012) and proposed the following seven categories of computational thinking concepts: (1) Abstraction and Problem Decomposition, (2) Parallelism, (3) Logical Thinking, (4) Synchronization, (5) Flow Control, (6) User Interactivity, and (7) Data Representation. Weese and Feldhausen (2017) and Feldhausen, et al. (2018) proposed the following concepts for the Computational Thinking with a focus on the concepts related to Computer Science Principles (see [Table 1](#)).

**Table 1.** Computational Thinking Concepts according to (Weese & Feldhausen, 2017)

---

**Computational Thinking Concepts**

---

**ALG** Algorithmic thinking – sequence of steps that complete a task. Operators and expressions are also included.

**ABS** Abstraction – generalized representation of a complex problem, ignoring extraneous information.

**DEC** Problem decomposition – breaking a problem into smaller, more manageable parts that can be solved independently of each other.

**DAT** Data – collection, representation, and analysis of data.

**PAR** Parallelization – simultaneous processing of a task.

**CON** Control flow – directs an algorithm's steps when to complete.

Scientific knowledge is embedded in social and cultural context.

**IAI** Incremental and iterative – building small parts of the program at each step instead of the whole program at once.

**TAD** Testing and Debugging – performing intermediate testing and fixing problems while developing.

**QUE** Questioning – working to understand each part of the code instead of using code that is not understood well.

---

Dede, Mishra and Voogt (2013) proposed the following computational practices:

**A. Abstracting:** “Computational thinking requires understanding and applying abstraction at multiple levels ranging from privacy in social networking applications, to logic gates and bits, to the human genome project, and more. Students use abstraction to develop models and simulations of natural and artificial phenomena, use them to make predictions about the world, and analyze their efficacy and validity. Students are expected to: Explain how data, information, or knowledge are represented for computational use; explain how abstractions are used in computation or modeling; Identify abstractions; and describe modeling in a computational context”.

**B. Analyzing problems and artifacts:** “The results and artifacts of computation, and the computational techniques and strategies that generate them, can be understood both intrinsically for what they are as well as for what they produce. They can also be analyzed and evaluated by applying aesthetic,

mathematical, pragmatic, and other criteria. Students design and produce solutions, models, and artifacts, and they evaluate and analyze their own computational work as well as the computational work that others have produced. Students are expected to: Evaluate a proposed solution to a problem; locate and correct errors; explain how an artifact functions; and justify appropriateness and correctness”.

**C. Communicating:** “Students describe computation and the impact of technology and computation, explain and justify the design and appropriateness of their computational choices, and analyze and describe both computational artifacts and the results or behaviors of such artifacts. Communication includes written and oral descriptions supported by graphs, visualizations, and computational analysis. Students are expected to: Explain the meaning of a result in context; Describe computation with accurate and precise language, notation, or visualizations; and summarize the purpose of a computational artifact”.

**D. Collaborating:** “Students can collaborate in a number of activities, including investigation of questions using data sets and in the production of computational artifacts. Students are expected to: Collaborate with another student in solving a computational problem; Collaborate with another student in producing an artifact; and collaborate at a large scale.

Bundy (2007) emphasize that “the ability to think computationally is essential to conceptual understanding in every field, through the processes of problem solving and algorithmic thinking”. NRC (2010) speaks about concepts from Computer Science, while many researchers put an emphasis on the fact that “Despite the obvious relevance of CT to computer science, scholars argue that CT needs to be taught in disciplines outside of computer science beginning in kindergarten” (Kotsopoulos et al., 2017; Barr & Stephenson 2011; Yadav et al., 2011). Abstraction is very fundamental concept and is closely related to the development of models, necessary for the Computational Pedagogy model which will be presented later. Two kinds of abstractions can be distinguished (OpenLearn from The Open University, n.d. a):

1. **Abstraction Modelling:** Abstraction as modeling can be understood in terms of the relationship between a part of reality and a model which represents the details of interest of this reality. For this reason, models are sometimes also referred to as representations.
2. **Abstraction as Encapsulation:** Abstraction as encapsulation involves two layers: The layer through which the user interacts with the model and is called the interface. This layer hides the detailed workings of the model from the user. The interface sits between the user and the layer at which the model is implemented. The latter is responsible for making the model do what it is supposed to do. This is where the automation of the model takes place. (OpenLearn from The Open University, n.d. a).

Although researchers have accepted that abstraction is a central concept in computational thinking, they are quick to disagree on the meaning of it (Cetin & Dubinsyb, 2017). Piaget “introduced the concept of reflective abstraction to describe the children’s construction of abstract logico-mathematical structures (Beth & Piaget, 1966) and he distinguished three types of abstraction: empirical, pseudo-empirical, and reflective abstraction.

Cetin & Dubinsyb (2017) assert that reflective abstraction can be used as a tool in the study of computational thinking by stating that “The most common meaning of abstraction of a concept in computer science and mathematics, is extraction, that is, the idea of considering common features of several examples and building a structure or category which has all of these features”. Wing (2008), connected abstraction with automation and she argued that the “mechanization of abstraction layers and the relationships between them leads to abstraction, and she defined that computing is the “automation of our abstractions”.

### **The Concepts of Computing-Computation-Computational**

“There is a lot of research about CT and computing and computation. Research papers use sometimes these concepts, as they are similar while some others differentiate them (implicitly or explicitly). We will try to delineate these terms, since their concise definitions will help us to proceed towards the Computational Science Education” (Psycharis, 2018). According to Wing (2008) “computing is the field that encompasses computer science, computer engineering, communications, and information science and information technology”. According to Denning (2003), “the principles of computing include: computation, communication, coordination, recollection, automation, evaluation and design and we can easily recognize that computing is connected to engineering design”.

The document by Computing at School Working Group (2012) entitled “Computer Science: A Curriculum for Schools” stresses that the working group recognizes that “Computer Science (CS) and Information Technology (IT) are disciplines within Computing that, like math’s or history, every pupil should meet at school.” In the same document there is a very precise terminology regarding computer science as covering principles such as algorithms, data structures, programming, systems architecture, design and problem-solving. We observe that there is, yet, no widely agreed definition of computing, and there is also no agreed definition for the Computer Science, either. For example, (Zendler & Spannagel, 2008), following a cluster analysis research study, state that “computer science includes the following central concepts: problem, data, computer, test, algorithm, process, system, information, language, communication, software, program, computation, structure, and model”.

Denning (2003) posited “computer science consists of mechanics (computation, communication, coordination, automation, and recollection), design principles (simplicity, performance, reliability and security) and practices (programming, engineering systems, modeling and validation, innovating, and applying)”. The term computation also appears in research papers. For example,

(Jona et al., 2014) state that “Computation is an indispensable component of STEM disciplines as they are practiced in the professional world. In the last twenty years, nearly every STEM field has seen the birth or reconceptualization of a computational counterpart, from Computational Engineering and Bioinformatics to Chemo metrics and Neuroinformatics”. In this article, we notice that computation is related to computational. Chande (2015) considers that one of the principles of computing is computation.

The term “computational” reinforces the confusion about these terms. For example, according to Wing (2008), a mathematical model is an abstraction of the physical-world phenomenon of interest. It ignores irrelevant detail and serves as a departure for automation. A computational problem is a problem that is expressed sufficiently precisely that it is possible to attempt to build an algorithm to solve it. In other words, Wing’s mathematical model is linked to a computational problem (OpenLearn from The Open University, n.d. a). “A computational problem as a problem that is expressed sufficiently precisely that it is possible to attempt to build an algorithm to solve it. In other words, Wing’s mathematical model is nothing other than a computational problem. Wing’s physical-world phenomenon is abstracted by the mathematical model, just like a computational problem provides an abstraction of a real-world problem. Algorithms and data structures form the link between a computational problem and its automation” (OpenLearn from The Open University, n.d. b).

In this framework computational includes computing and computation, leading to the solution of the computational problem using models that will be simulated. Pedaste & Palts (2017) discuss the concept of computational learning as an iterative and interactive process (between the student and the model of computation). Later, when we will discuss the engineering design, we will notice that iterative and interactive process are also mentioned in the engineering design (Katehi, et al., 2009). From the brief analysis presented above, it is evident that the terms computing, computation, computational are used sometimes with the same meaning (i.e. algorithms, make calculations etc.) and in other cases computational means something wider than computing.

### **CT and Science-Engineering and Mathematics**

In this section, we will explore the relation between CT and the cognitive areas included in STEM. Zhang and Luo (2012) suggest the integration of CT with Mathematical Thinking, Science thinking and Engineering Thinking. According to (Sengupta et al., 2013) CT encompasses being able to distinguish several levels of abstraction and apply mathematical reasoning and design-based thinking. According to (Weintrop et al., 2015), Science and Mathematics are becoming computational endeavors and we have to use computational methods. Next Generation Science Standards (NGSS, 2013) also suggest that CT “is a core scientific practice and due to the increased presence of computation in mathematics and scientific contexts, a new urgency has come to the challenge of defining computational thinking and providing a theoretical grounding for what form it should take in Science and Mathematics”.

“Bringing computational tools and practices into mathematics and science classrooms gives learners a more realistic view of what these fields are, better prepares students for pursuing careers in these disciplines and from a pedagogical perspective, the thoughtful use of computational tools and skill sets can deepen learning of mathematics and science content” (e.g. National Research Council 2011a, b). Next Generation Science Standards (2013) includes computational thinking as a core scientific emphasizing the necessity of computation in teaching and learning practices. (Weintrop et al., 2016) proposed a taxonomy of computational thinking in Math’s and Science consisting of four main categories: data practices, modeling and simulation practices, computational problem-solving practices and systems thinking practices. These practices include—among others—the collection and analysis of data, the design, construction and test of computational models, the assessment of different solutions to solve a problem, the management of the complexity of a system and the investigation of a system as a whole. These practices are strongly related to the CT dimensions and we should extend them to include engineering design, as is also suggested by (Katehi et al., 2009). During this article we will try to justify the model of “Computational STEAM content Pedagogy” by using arguments based also on these practices.

## Computational Science in Education (CSE)

### Introduction

We have reported that the term “computational” includes “computing” and “computation”. We consider that “computational” is equivalent to the development of a model -through an abstraction- that will be simulated –according to a method of simulation- and will produce data that will be collected, analyzed and tested against the real data. “Computational” can be a method which implements Computational Thinking as a real experiment, as suggested by research in order to solve a real/authentic problem. For example, Bundy (2007) posited “the ability to think computationally is essential to conceptual understanding in every field, through the processes of problem solving and algorithmic thinking”. In the last twenty years, nearly every STEM field has seen the birth or reconceptualization of a computational counterpart, from Computational Engineering and Bioinformatics to Computational Psychology, Neuroinformatic etc. Jona et al., (2014) state that “one of the fundamental research questions in the STEM agenda is how can we increase computational competencies for all students and build interest in computing as a field in its own right?”

According to Weintrop et al. (2016), “bringing computational tools and practices into Mathematics and Science classrooms gives learners a more realistic view of what these fields are, better prepares students for pursuing careers in these disciplines and from a pedagogical perspective, the thoughtful use of computational tools and skill sets can deepen learning of mathematics and science content” (e.g. National Research Council 2011a, b).

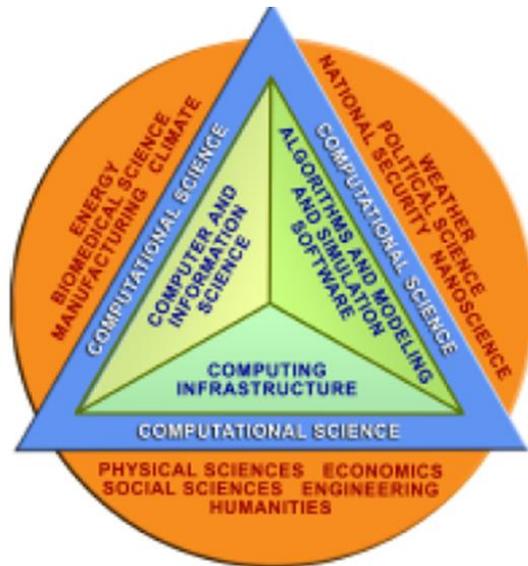
As stated in Barr & Stephenson (2011), “Computer Science is related to Computational processes and scientists can advance understanding of how to bring computational processes to solve problems in other fields and on problems that lie at the intersection of disciplines. For example,

bioinformatics and computational biology are different, but both benefit from the combination of biology and computer science. The former involves collecting and analyzing biological information, the latter involves simulating biological systems and processes”.

The concepts of computing and computational appear in (Yasar et al., 2016). Authors state that Computational Pedagogy is an inherent outcome of Computing, Math, Science and Technology integration. In the same article, Computing is related to algorithmic and programming. They also suggest that computational modeling and simulation technology (CMST) can be used to improve technological pedagogical content knowledge (TPACK) of teachers. Chande (2015) states that “the science that scientists and researchers developed drawing inspirations from natural processes now looks to be taking the center stage and reversely motivating them to decipher natural processes as computational activities.” Bienkowski et al. (2015), assert that “projects with an orientation to computational science tend to emphasize data, modeling, and systems thinking”. In this perspective the term “Computational Science” has a lot of similarities with the four taxonomies of (Weintrop et al., 2016) mentioned before.

The relation between CT and “Computational” is emphasized by Aho also (2012) who defined “CT as the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms”. Computational Science (CS), in general, has its origins in Monte Carlo modeling and algorithms like Lanczos algorithm, for applications of stochastic statistical sampling for solving complex problems in Physics (Landau et al., 2008; Psycharis, 2016a,b). Computational Science (CS) is the integration of Mathematics, Computer Science and any other discipline to explore authentic-complex problems. It brings together concepts from a variety of cognitive subjects (Landau et al., 2008) and is considered to be part of the Computational Science-Engineering community. **Figure 1** presents the components of Computational Science and has been taken from Report to the President. Computational Science: Ensuring America’s Competiveness (2005). Wolfram (2002) proclaimed the emergence of “a new kind of science based on computational experiments into emergent patterns in nature, arguing such explorations are not possible without computation. Scientific fields are undergoing a renaissance in experimental approaches primarily due to the availability of more powerful computers, accessibility of new analytical methods, and the development of highly detailed computational models in which a diverse array of components and mechanisms can be incorporated”.

Although some computer Science and Mathematics programs have championed this new field, Computational Science, also finds strong allies in other disciplines, particularly Physics and Biology. Computational Science and Computer Science have common concerns:



**Figure 1.** Components of Computational Science

1. *When it comes to computer performance and application optimization;*
2. *Computational Science and Mathematics have common concerns when it comes to applied math techniques.*

Initially Computational Science was considered as a bridge between different disciplines but after the first phase (recognition phase); this area developed its own methods. According to (Yaşar, 2004; Yaşar, 2013; Yaşar & Landau, 2003) Computational Science (CS) overlaps with many other knowledge areas, so an educational program in Computational Science, naturally draws strength from all of them. Nevertheless, in addition to over-lapping with Computer Science, Math, and Science and Engineering application areas, Computational Science has its own core knowledge area. Juszczak (2015) states that Computational Science, in both natural and social sciences, is different from the usage of computers to analyze complex systems and data sets. Computational Science is a non-empirical science. Data that is gathered in Computational Science is the result of simulations and virtual experiments. The key distinction between a true Computational Science and a Science that uses computation is in the nature of evidence: traditional science and science experimentation use computation to assist in the analytic and experimental process have, as their threshold of truth, empirical evidence. Computational Science, on the other hand, conducts experiments that are only virtually true and attempts to use data about the real world in order to conduct real experiments in a virtual universe.

### **Modelling in Education**

An essential component of the Computational Science (CSE) is the development of models. Models could be explorative or/and expressive, but in any case they should be connected to modeling indicators. Hestenes (1999) states that most physics and generally science/engineering problems are solved by constructing or selecting a model, from which the answer to the problem is extracted through model-based inference. In a profound sense the model provides the solution to the problem. Thus, an emphasis on models and modeling simplifies the problem and organizes a physics course into understandable units.

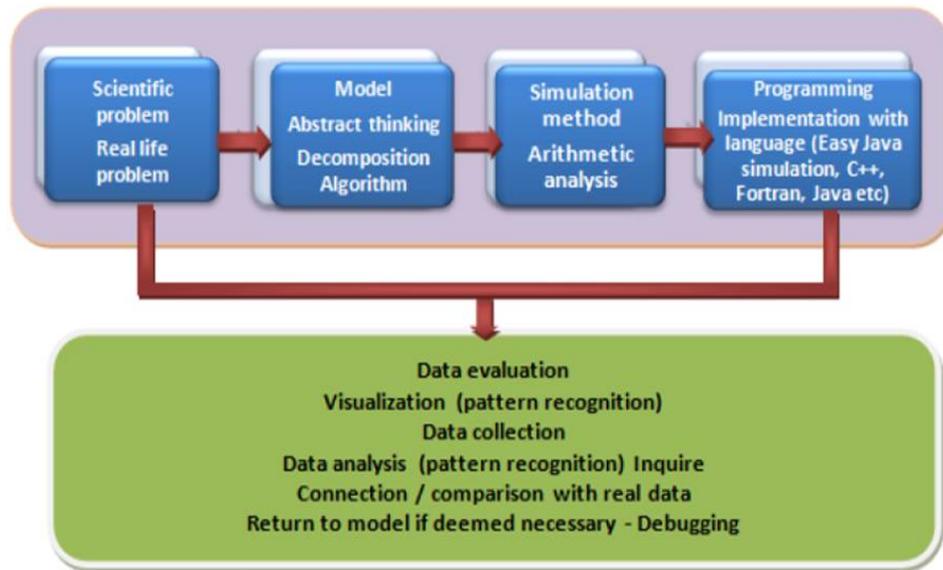
Models play a fundamental role in science and mathematics teaching (Mendoca & Justi, 2013). According to literature (see e.g. Justi & Gilbert, 2002; Psycharis, 2016a; Psycharis, 2016b) models are explanatory tools with predictive power. Justi and Gilbert (2002) claim that “when learning science, students learn about the nature, scope and limitations of the scientific or curriculum models; when learning about science, students learn to evaluate the role of models in the development and dissemination of the results of scientific research; and when doing science, students learn to elaborate, express, and test their own models. These interpretations suggest the relevance of the inclusion of modelling activities in science education as a way to promote authentic science learning” (Mendoca & Justi, 2013). We consider modelling as a central issue in the CSE methodology while the Computational experiment (see next section for full description of the Computational Experiment) implements CT in practice in accordance to research (see for example Bienkowski et al., 2015), where it is clearly stated that Computational science tend to emphasize data, modelling, and systems thinking.

### **The Computational Experiment**

One of the crucial components of C.S. is also the abstraction of a physical phenomenon to a conceptual model and its translation into a computational model that can be validated. This leads us to the notion of a computational experiment (CE), where the model and the computer take the place of the “classical” experimental set-up and where simulation(as working model) replaces the experiment (Psycharis, 2016a,b; Psycharis et al., 2018). CSE focuses on the form of an authentic problem to solve and follows a scientific problem-solving paradigm (Computational experiment –CSE-approach), with a sequence of steps as follows (see **Figure 2**):

- A. Problem** (from science/real world);
- B. Modeling** (Mathematical relations between selected variables-decomposition of the problem);
- C. Simulation Method** (time dependence of the state variables, discrete, continuous or stochastic processes, selection of proper interfaces);
- D. Development of the algorithm** based on numerical analysis methods;
- E. Implementation of the algorithm** (using Java, Scratch, Python, Arduino, raspberry pi etc.);

**F. Assessment and Visualization** through exploration of the results and comparison with real data received from authentic phenomena. CSE shares many commonalities with CT and may serve as the background platform to implement applications that include the dimensions of CT. The different steps of the CE are presented below:



**Figure 2.** The CE steps

### STEM Cognitive Areas

According to NAE (2014):

1. **Science** is “the study of the natural world, including the laws of nature associated with physics, chemistry, and biology and the treatment or application of facts, principles, concepts, or conventions associated with these disciplines. Science is both a body of knowledge that has been accumulated over time and a process—scientific inquiry—that generates new knowledge. Knowledge from science informs the engineering design process”.
2. **Technology**, “while not a discipline in the strictest sense, comprises the entire system of people and organizations, knowledge, processes, and devices that go into creating and operating technological artifacts, as well as the artifacts themselves. Throughout history, humans have created technology to satisfy their wants and needs. Much of modern technology is a product of science and engineering, and technological tools aroused in both fields”.

3. **Engineering** “is both a body of knowledge—about the design and creation of human-made products—and a process for solving problems. This process is design under constraint. One constraint in engineering design is the laws of nature, or science. Other constraints include time, money, available materials, ergonomics, environmental regulations, manufacturability, and reparability. Engineering utilizes concepts in science and mathematics as well as technological tools”.
4. **Mathematics** is “the study of patterns and relationships among quantities, numbers, and space. Unlike in science, where empirical evidence is sought to warrant or overthrow claims, claims in mathematics are warranted through logical arguments based on foundational assumptions. The logical arguments themselves are part of mathematics along with the claims. As in science, knowledge in mathematics continues to grow, but unlike in science, knowledge in mathematics is not overturned, unless the foundational assumptions are transformed. Specific conceptual categories of K–12 mathematics include numbers and arithmetic, algebra, functions, geometry, statistics, and probability. Mathematics is used in science, engineering, and technology”.

According to (Xie et al., 2011), the Simulation-based Engineering and Science (SBES) is considered as cognitive area that provides the support for the development of models in Mathematics and Physics that will be simulated. Development of models is based on mathematical reactions between the variables selected. Algorithms will be used next in order to implement the model using a programming language. This methodology is called “Computational Experiment Mathematical Modeling (CEMM)”. Kroes & Van de Poel (2009), propose two aspects for Technology:

1. Technology as *process/activity* which includes the collection of design processes, construction, and development of artifacts.
2. Technology as *product*, i.e. a collection of artifacts.

At the report entitled “Standards for Technological Literacy: Content for the Study of Technology” by the Committee for International Technology Education Association (ITEA, 2007) gives issues related to the content of “Education in Technology”, and five indicators/standards as follows:

- A. The Nature of Technology
- B. Technology and Society
- C. Design
- D. Abilities for a Technological World
- E. The Designed World

Consideration of questions like a number of specific questions, such as “How does engineering education “interact” with Science, Technology, and Mathematics?” Scientists “discover” knowledge and Engineers use this knowledge to “produce” operational “products” (Carlson & Sullivan, 2004). Scientific research creates data that will be taken account by engineers in order to proceed to the engineering design. Engineering design can be used in a context around a real problem which can be used for conceptual understanding of scientific concepts. In education settings it is better to think between science and engineering in a dual relationship. To make the link between the cognitive areas included in STEM more concise, NGSS (2013) suggests that STEM education should focus on:

1. Scientific and engineering practices
2. Crosscutting concepts that unify the study of science and engineering through their common application across fields
3. Core ideas in four disciplinary areas: physical sciences; life sciences; earth and space sciences; and engineering, technology, and applications of science

These are described below:

#### A. Scientific and Engineering Practices

1. Asking questions (for science) and defining problems (for engineering)
2. Developing and using models
3. Planning and carrying out investigations
4. Analyzing and interpreting data
5. Using mathematics and computational thinking
6. Constructing explanations (for science) and designing solutions (for engineering)
7. Engaging in argument from evidence
8. Obtaining, evaluating, and communicating information

#### B. Crosscutting Concepts

1. Patterns
2. Cause and effect: Mechanism and explanation
3. Scale, proportion, and quantity
4. Systems and system models
5. Energy and matter: Flows, cycles, and conservation
6. Structure and function
7. Stability and change

#### C. Disciplinary Core Ideas (we present these only for physical sciences)

Physical Sciences

PS1: Matter and its interactions

PS2: Motion and stability: Forces and interactions

PS3: Energy

PS4: Waves and their applications in technologies for information transfer

The above framework has as main objective to “actively engage students in scientific and engineering practices and apply crosscutting concepts to deepen their understanding of the core ideas in these fields”(NGSS,2013).

### Epistemology of STEM Cognitive Areas

We will analyze the epistemological content of STEM disciplines with focus on their educational content. According to (Chandler et al., 2011) an epistemology is a way of reasoning and understanding the things we encounter in the world. An education scenario should be taken into account the epistemology, according to the following:

1. **An epistemology** comprising a theory of the nature, genesis, and warranting of subjective knowledge, including a theory of individual learning as well as a theory of “truth.”
2. **A methodology**: a theory of which methods and techniques are appropriate and valid to use to generate and justify knowledge, given the epistemology.
3. **A pedagogy**: a theory of teaching, the means to facilitate learning according to the epistemology”. (Gunawardena Egodawatte, PhD Thesis, Secondary School students’ misconceptions in Algebra, University of Toronto, 2011).

#### *Science Epistemology for Education*

A fundamental issue in the Epistemology of Science (Life sciences, Physical Sciences, Earth and Space Sciences, and Applied Sciences) is the concept of “evidence based”, which is supported by the collection and analysis of data and the argumentation(Ates& Cataloglu,2007; Lawson et al., 2007;Psycharis,2013). According to (NGSS,2013) Science education should focus on the “core ideas in the disciplines of Science as well as crosscutting ideas such as mathematization, causal reasoning, evaluating and using evidence, argumentation, and model development and how crosscutting ideas may play out in the context of select core disciplinary ideas and articulate expectations for students’ learning. Science epistemology should be implemented in school education by engaging students in scientific practices, in applicability of crosscutting concepts across science disciplines and in the exploration of the relation of scientific concepts to Engineering and Technology. The term “practices” is used instead of the term “skills” to emphasize that engaging in scientific investigation requires not only skill but also knowledge that is specific to each practice (NGSS, 2013). Scientific practices are presented below (NGSS, 2013) and are related to the Inquiry Based Teaching and Learning approach (Assay & Orgill, 2010):

1. *Asking questions (for science) and defining problems (for engineering)*
2. *Developing and using models*
3. *Planning and carrying out investigations*
4. *Analyzing and interpreting data*
5. *Using mathematics and computational thinking*
6. *Constructing explanations (for science) and designing solutions (for engineering)*
7. *Engaging in argument from evidence*
8. *Obtaining, evaluating, and communicating information*

Inquiry based teaching and learning has officially been promoted as a pedagogy for improving science learning in many countries (Bybee, Trowbridge & Powell, 2008) and can be defined as “the deliberate process of diagnosing problems, planning experiments, distinguishing alternatives, setting up investigations, researching conjectures, sharing information, constructing models, forming coherent arguments, collecting and analyzing data” (Bell, Hoadley & Linn, 2004; Bell et al., 2010; Assay & Orgill, 2010; Bybbe et al., 2008; Psycharis, 2016a, 2016b). The nine inquiry tools of Bell et al. (2010) are closely related to the essential features of Inquiry (Assay & Orgill, 2010), namely:

- Question (the learner engages in scientifically oriented questions),
- Evidence (the learner gives priority to evidence),
- Analyze (the learner analyses evidence),
- Explain (the learner formulates explanations from evidence),
- Connect (the learner connects explanations to scientific knowledge) and
- Communicate (the learner communicates and justifies explanations).

Significant parts of scientific research are carried out on models rather than on the real phenomena because by studying a model we can discover features of and ascertain facts about the system the model stands for (Swoyer, 1991). This cognitive function of models has been widely recognized in the literature, and some researchers even suggest that models give rise to a new form of reasoning, the so-called ‘model based reasoning’ (Magnani & Nersessian, 2002) while modeling ability is closely associated to model-based reasoning (Chittleborough & Treagust, 2007). It is well known that scientific theories developed through a process of continuous elaboration and modification in which scientific models are developed and transformed to account for new phenomena that are uncovered in exploring a knowledge area. Similar processes are involved in students’ learning of scientific concepts when students develop conceptual models (e.g. Bell et al., 2010; Nersessian, 1992; White, Frederiksen & Spoechr, 1993). In a similar fashion, Inquiry based learning requires from students to make successive refinements to their mental models in order to transform them to conceptual models that align to scientific theories. In this context, models are considered as pedagogical tools that have the potential to drive changes in the approaches to learning, while they can help students to develop coherent conceptual models (Justi & Gilbert, 2002).

### **Computational Experiment (CE) and Inquiry Based Teaching and Learning approach**

In order to describe discovery/inquiry based learning as a research process, Shunn and Klahr (1995) and Klahr and Dunbar (1998) introduced the hypothesis and the experimental spaces in order to describe the discovery/inquiry based learning as a search process. In their model, the hypothesis space contains all rules and variables describing the specific domain, while the experiment space consists of all experiments that can be implemented within this domain.

Psycharis (2011) extended these spaces in order to include the computational experiment approach and suggested three spaces for the computational experiment, namely:

1. **The hypotheses space**, where the students in cooperation with the teacher, decide, clarify and state the hypotheses of the problem to be studied, as well as the variables and the concepts to be used as well as the relations between the variables.
2. **The experimental space**, which includes the model and the simulation for the problems under study. In this space the learners are engaged in the scientific method writing models according to the interaction laws that govern the phenomenon.
3. **The prediction space**, where the results, conclusions or solutions formulated in the experimental space, are checked through the analytical (mathematical) solution as well as with data from the real world”.

We can notice that the role of modeling is essential both as an inquiry tool and as a feature of inquiry. The fundamental component of the Computational Experiment (CE) is the development of models, which makes it strongly connected to inquiry processes. Our proposal integrates inquiry approach and Computational Experiment (CE) through the interconnection of the CE spaces, namely the hypotheses space, the experimental space and the prediction space with the essential features of inquiry and the inquiry tools (Psycharis, 2016a, Psycharis, 2016b).

In **Table 2** below, we present the relation between the spaces of the CE and the features and the tools of inquiry:

**Table 2.** The relation between the spaces of the CE and the features and the tools of inquiry

Spaces of the Computational Experiment (Psycharis, 2013)	Essential Features of Inquiry (Assay & Orgill, 2010)	Inquiry tools of (Bell et.al, 2010)
Hypotheses space	Question	Orienting and asking questions; generating hypotheses
Experimental space	Evidence Analyze	Experimental space
Prediction Space	Connect	Prediction Space

### **Epistemology of Technology**

Mitcham (1994) distinguished four scopes of Technology: namely technology as (1) object, (2) knowledge, (3) activity, and (4) volition. According to the International Technology and Engineering Educators Association (ITEEA) Technology includes: “innovation, change, or modification of the natural environment in order to satisfy perceived human wants and needs”. Thus, a technologically literate person is able to:

1. **Use technology:** successful operation of key products & systems of the time; knowing components of existing macro-systems, or human adaptive systems, and how the system behave;

2. **Manage technology:** ensuring that all technological activities are efficient and appropriate;
3. **Evaluate technology:** being able to make judgments and decisions about technology on an informed basis rather than on an emotional one
4. **Understand technology:** more than knowing facts and information, but also the ability to synthesize the information into new insights” (Goris & Dyrenfurth, 2010)

### *Engineering Education Epistemology*

Researchers and professional associations provide further compelling rationales for inclusion of engineering in K-12 curriculum, either as a course in its own right or woven into existing mathematics and science courses. Some of these rationales for the inclusion of engineering in K-12 coursework include the following “(Brophy et al., 2008; Hirsch, Carpinelli, Kimmel, Rockland & Bloom, 2007; Koszalka, Wu & Davidson, 2007):

1. Engineering provides a real-world context for learning mathematics and science;
2. Engineering design tasks provide a context for developing problem-solving skills; and
3. Engineering design tasks are complex, and as such, promote the development of communication skills and teamwork” (Katexi et al., 2009)

According to (Cross, 2017), “Engineering design is the process of developing a concrete solution for an ill-defined problem within technical feasibility constraints.” Below five types of engineering skills, are presented:

- A. **Intentional designing:** Look for pre-planning or evidence of thinking ahead.
- B. **Analyzing:** Look for novel or risky efforts.
- C. **Refining and testing:** Look for efforts that repeat and improve each time in order to reach a goal.
- D. **Prototyping:** Look for evidence of a student modeling an idea to reach a goal.
- E. **Communicating design:** Look for students sharing ideas about something they’re planning or creating” (Hellow Worls, Issue 5, 2018).

### *Epistemology of Mathematics Education*

In the perspective of Piaget’s epistemology (Piaget, 1970), learning mathematics is seen as a continuous process through abstraction of relationships between actions and reflections. During this process, students construct schemas and modify and/or apply them intentionally to achieve their goals. Careful analysis of these actions will allow the researcher or the teacher to identify student conceptions or misconceptions wherever they may lead (Egodawatte, 2011). Mathematics epistemology is affected by the use of computers. According to (Nievergelt, 1974) “when a mathematical problem is approached with a view towards using a computer in its solution, the situation is very similar to that of an experimental scientist in a laboratory experiment”. According

to Hußmann, et al. (2019) “an epistemological theory in mathematics education offers the opportunity to trace students’ conceptual development in both its individual and social facets through analyzing patterns of reasoning and can be used to illustrate the relationship between mathematical standard and individual ways of reasoning in conceptual development processes.”

## **STEM Epistemology**

### *The Interdisciplinary and Transdisciplinary Epistemology of STEM Education*

There are different concepts and views about these terms and we will try not to present all the views but rather we will present some views that are closer to our purpose to connect epistemology with STEM. According to Toomey et al. (2015) “Interdisciplinarity is not just research in two or more different disciplines, nor is it adding methodologies from other disciplines to an already discrete project; rather, it is an integrated approach to answering a question that recognizes the limitations inherent in the compartmentalized system of academic research. Transdisciplinary work moves beyond the bridging of divides within academia to engaging directly with the production and use of knowledge outside of the academy.” Psycharis (2018) also asserts that “STEM epistemology needs to integrate knowledge, perspectives and interests not only from different disciplines but also from related societal actors when the research project is designed. We consider that this issue as very important for a real STEAM epistemology implementation as societal factors should inspire STEM research in school settings and in Higher Education new curricula as well as for the creative industry.” While transdisciplinary is defined as “concerned with creating new, integrative knowledge to address the complex problems of the world. (Antola et al., 2013) demonstrate examples of pedagogy and learning using —transdisciplinary approaches which involve multiple disciplines and the space between the disciplines with the possibility of new perspectives 'beyond' those disciplines” (Nicolescu, 2002; McGregor, 2015, as cited in Psycharis, 2018).

In summary, we consider that in Interdisciplinary approach there is a transfer of methodologies from one discipline to another to address a problem and this sometime leads to a new discipline (e.g. bioinformatics). In transdisciplinary new approaches are created and integrated while considering complex social issues. Borrego & Newswander (2008) state that “in a truly interdisciplinary approach to collaboration, researchers from different disciplines work in a more integrated way to solve a problem together. Rather than each contributing separate pieces to the solution, the collaborators work closely together, combining their knowledge from their own disciplines to work toward a solution. At the end of a truly interdisciplinary collaboration, each collaborator is changed by the experience.” There is an important feature of interdisciplinary education that can be best described by Aristotle’s well known statement, “the whole is more than the sum of its parts,” or the theory of Gestalt psychology, “the whole is other than the sum of its parts,” which means that the whole has a reality of its own, independent of the parts (Yaşar et al., 2015).

### *The approaches to STEM Education*

There are two approaches for STEM education integration: the content integration and the context integration. These different approaches allow teachers flexibility on how they integrate STEM in their classrooms. Content integration (Moore, 2008) focuses on the merging of the content fields into a single curricular activity or unit to highlight big/crosscutting ideas from multiple content areas. Consider for example the operation of wind turbines to illustrate the power and possibilities of teaching within a fully integrated STEM context. The wind turbine design lessons utilize robust hands-on wind turbine kits that allow teachers and students to explore the variables that impact electricity generation. Teachers can direct engineering design by considering a model construction (or they can ask students to create the model) selecting the variables of the phenomenon and the relation between the variables.

Engineering design is included by designing a prototype according to the scientific concepts included and by asking questions about the material, shape, length etc. of the blades introducing crosscutting ideas like “Cause and effect, Mechanism and explanation, Energy and matter: Flows, cycles, and conservation”. A full understanding of an optimal wind turbine design also involves developing and applying physics concepts related to electricity generation, the mathematical concepts (related to trigonometry, rotation, and gear system). This STEM curriculum activity needs a series of lectures to be implemented and faces a problem of real life. It is usual this problem to be faced as a whole and not in separate issues (i.e. first discussing issues from physics, next move to mathematics etc.). A unit such this allows a teacher to teach concepts from each discipline and highlight how these disciplines are all needed to solve a problem in this area. In this example, students can design and make their artifact, test this against the experimental data and reframe their considerations about the prototype. This process can be implemented either by using the Computational Experiment or using physical computing (e.g. Arduino construction), or without using computers, i.e. unplugged computing. You can find also a very interesting example in the article of (Schmittka et al., 2010). In the STEM context integration approach, the focus is on the content of one discipline and next contexts from other disciplines are used to make the content more relevant.

For example, a mathematics teacher might choose a unit from probability about Bayes theorem and then he can ask students to analyse samples from a biochemistry lab in order to examine the probability for diseases using conditional probabilities. In another example, teacher teaches algorithms and then ask engineering students to visit different networks and register the response time in a network with different number of nodes. We have observed a lot of confuse about the terms that describe STEM epistemology, which led to different integrations in Education (primary, secondary and higher). Similar thoughts are shared by NAE and NRC report (2014) where it is stated that in educational practice and in research, the term integrated is used loosely and is typically not carefully distinguished from related terms such as connected, unified, interdisciplinary, multidisciplinary, cross-disciplinary, or transdisciplinary. Our suggestion is to

adopt a STEM epistemology that is in alignment with the Mode-2 approach as discussed by Nicolescu (2004).

We claim that STEM epistemology is closely related to Mode-2 system as it faces problems that emerge from different disciplines and loose organizational structures, flat hierarchies, and open-ended chains of command are dominant. Complexity according to Nicolescuian methodological approach, should also be related to STEM content epistemology. According to (Nicolescu, 2004), complexity is a modern form of the ancient principle of universal interdependence, in that everything is dependent on everything else, everything is connected, and nothing is separate. This definition of complexity, alongside with current research efforts to define complexity, raise awareness about issues like emerging behavior, connection of scales etc. that could be related to STEM content, as STEM faces complex problems. Our suggestion is based on the consideration that the whole is qualitatively different from its parts and this panoramic view sometimes focuses on specific discipline (STEM context approach) but in general moves with a holistic way between the disciplines. This approach put an emphasis on the whole and on the correlation of concepts and phenomena and not on the separate phenomena enhancing the abstraction skills as well as the modeling practices of the Computational Thinking. Succinctly, we propose a definition of STEM as a holistic STEM content approach which follows the interdisciplinary approach and can be implemented using the Computational Science and the Inquiry teaching and learning approach.

### *Art-science-technology integration*

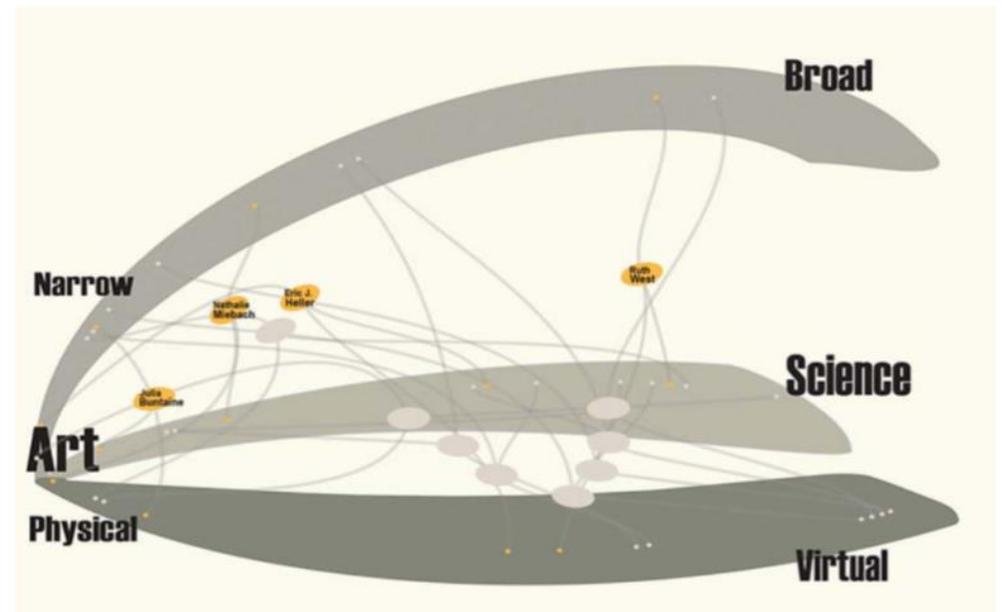
The basic skills for the Education and society involve: a) Problem skills for complex problems, related to complexity, b). Critical Thinking and Divergent Thinking and c). Creativity (ambiguity and uncertainty for ill-defined complex problems). In interdisciplinary approach there are chances for generating new knowledge which lies in between disciplinary boundaries integrating knowledge to address the complex problems of the world. In such approaches which involve multiple disciplines the interaction of methodologies in the space between and at the intersection between the disciplines offer the possibility of new perspectives 'beyond' those disciplines. We will try to justify the intersection between STEM disciplines and Art and more specifically the relationship of Arts with Engineering design. Art and Science –Engineering have a common history and “there has been some debate and research that suggests the arts are well-suited to be combined with science, technology, engineering, and math disciplines making the STEM acronym STEAM” and interconnectivity between the arts and sciences is an area of research and practice that can be traced throughout history (Ghanbari, 2015).

According to Daugherty (2013) “Modern cell phones and PDA’s use a form of encryption called frequency hopping to ensure your messages cannot easily be intercepted. Frequency hopping was invented by the composer George Antheil in collaboration with the actress Hedy Lamarr and Computer chips are made using a combination of three classic artistic inventions: etching, silk screen printing and photolithography. Eisner and Powell (2002) also questioned the notion that

art and science belong in different worlds, and noted synergies across the different disciplines. The STEM to STEAM movement presents new language to frame such interdisciplinary thinking.” Mishra and Yadav (2013) have argued that human creativity can be augmented by computational thinking, which could move students from being consumers of technology to create new forms of expression build tools and foster creativity an essential characteristic of Art. “The creative process doesn’t exist in a vacuum—it’s a highly integrated activity reflecting history, aesthetic theory, and often the technological breakthroughs of the day. This was certainly the case during the Renaissance, when artists, engineers, scientists, and thinkers all came together to create truly remarkable works of art and engineering” (Ira Greenberg, *Processing Creative Coding and Computational Art*, 2007). Arts integration is connected with the didactic strategy of Inquiry based teaching and learning approach. Ghanbari (2015) states that “While it is not the primary role of the arts in academia, visual and performing arts have the ability to enhance learning in other subjects. A large facet of arts coursework is inquiry-based, which means it revolves around questioning and understanding concepts versus finding the answer to a given problem.”

According to Greenberg (2007), even programming can be learned very easily through the creation of screen art by using the processing language. In an effort to help us frame an understanding of the diverse and complex nature of **art-sci-tech** collaborations (Campbell & Samsel, 2015) put together a basic visual cognitive tool with which to facilitate thinking and build a dialogue (see **Figure 3**). “Art-sci-tech collaborations exist on a number of spectrums. Having an understanding of the spectrums will help provide a language with which to think about the following spectrum:

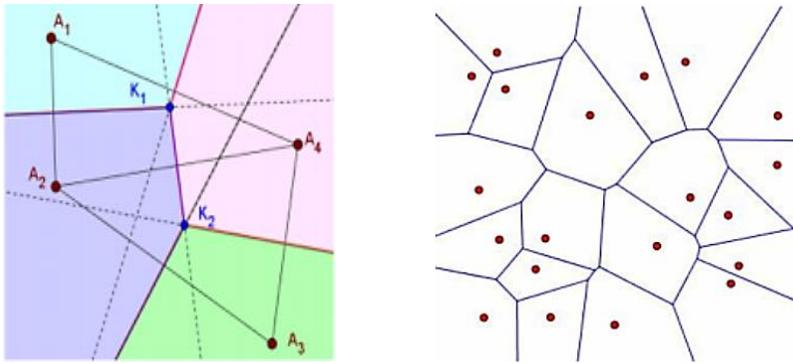
1. The first thing to consider is the intent of the work. Is work being presented as a work of art, a work of science, or some combination of both?
2. A second spectrum regards the breadth of the subject matter. Just as scientific research can be broad in scope, exploring wide reaching areas of understanding, so often is art. Also like science, art may instead be focused on a specific area of scientific research or revolve around a specific experiment.
3. As a third example of a continuum on which work can be considered, the physical-virtual continuum addresses the physical properties of the work. Is it a sculpture that has mass and sits on a pedestal, or is it an idea, differing by each person’s interpretation? Or is it something in between that resides on a disk, is broadcast via the airwaves, or can be realized physically according to some digital template?” Robotics is connected to Arts. (Martin et al., 2009) created a University project called Artbotics, which combined computing, robotics, and interactive arts, and all students were expected to be engaged in all these areas. Students engaged in hands-on work with robotics materials in the service of creating new media art.”



**Figure 3.** Creative tool for categorizing each artist’s work in a 3D space. (An interactive version is available on the Web at <http://bdcampbell.net/ieee/cga/>).

Psycharis (2019) previously asserted that, for an art-sci-tech implementation, teachers should possess the following qualifications: a) good knowledge of science ideas and capacity in technological concepts; b) experience in connecting Computational Thinking (mainly abstraction in different levels) with science and technology courses; c) willingness to apply crosscutting (transversal/crosscutting ideas) in courses in alignment with Disciplinary Core Ideas; and d) capacity to connect scientific concepts with real life phenomena. In addition, experts should promote the development of curricula that contain examples of use of e.g. concepts from engineering in arts.

For example, consider the well-known Voronoi diagrams. Voronoi diagrams are created using simple mathematical formulas and they can be used in many areas of science and technology. Voronoi diagrams were invented by the Russian Mathematician Georgi Voronoi (1868-1908) and they were used by John Snow to justify the expansion of cholera in London. (<https://plus.maths.org/content/uncoveringcause-cholera>). Given  $n$  points we can split the spaces in  $n$  different regions using the vertical at the middle of each segment formed by two points. The space is divided in  $n$  different regions (see **Figure 4**). The region of each points is formed by the points that are closest to this point. (e.g. Barequet et al., 2002) while Voronoi diagrams have been used in many disciplines (e.g. Okabe,1992).



**Figure 4.** The Voronoi diagrams with 20 points (<http://www.ams.org/publicoutreach/feature-column/fcarc-voronoi>)

Science and Arts are linked in many perspectives. For example solar Equation is a large-scale public art installation that consists of a faithful simulation of the Sun, scaled 100 million times smaller than the real thing. The solar animation is constructed by live mathematical equations that simulated the turbulence and flames that can be observed on the surface of the Sun ([http://www.lozano-hemmer.com/artworks/solar\\_equation.php](http://www.lozano-hemmer.com/artworks/solar_equation.php)).

#### *Emotional intelligence and Physical computing*

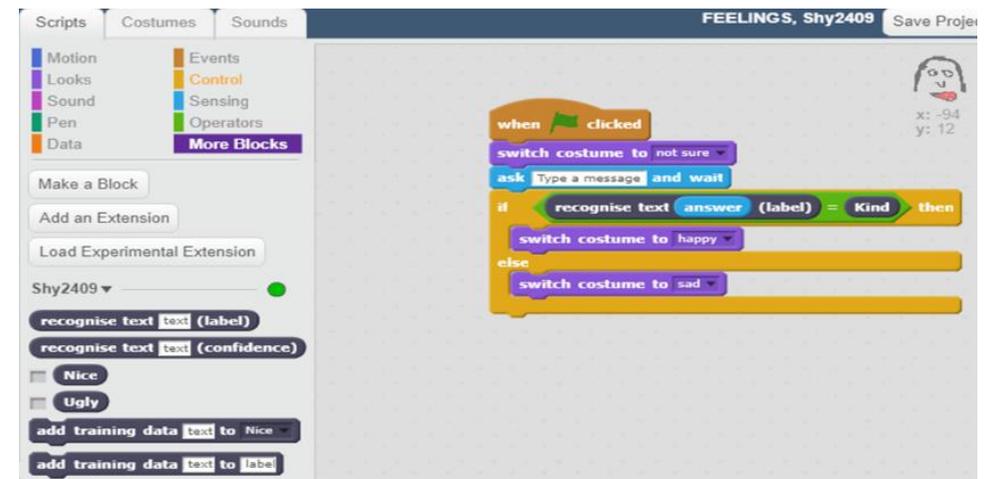
Physical computing is considered as a linkage between the computers to the physical/real world (Libow Martinez & Stager, 2013) and as a suitable means a suitable candidate in order to combine digital elements with the real world, as it is about creating an interface conversation between the physical world and the virtual world of the computer (Schulz & Pinkwart, 2015). In the context of computer science education most of the research focus on programming (e.g. Qiu et al., 2013, Psycharis et al, 2017). Physical computing can be implemented in computer science in two ways: either to teach computer science with physical computing, or to selectively use physical computing as an entry point to different topic areas of computer science (Przybylla & Romeike, 2014). Physical computing takes the computational concepts “out of the screen” and into the real world so that the student can interact with them by changing the model (Rubio et al., 2013). For example, we can develop a small artifact with cheap materials and inside this we can set the Arduino. The robot (see Figure 5) responds to the motion of a servo motor and moves its eyebrows according to its feelings (Kalovrektis and Psycharis, courses at Michel Cacoyiannis foundation, MCF, Athens, Greece).



**Figure 5.** A robot with emotions using simple materials and the Arduino platform.

#### *Emotional intelligence and Machine Learning*

Emotional intelligence can also be implemented using machine learning in education. Using the platform <https://machinelearningforkids.co.uk/> we can create emotions that respond to specific expressions. Using this platform the system learns some expressions and after training it recognizes expressions “similar” to the expressions written, with great confidence. Using the Scratch environment (see Figure 6) students create the code and they are involved in the development of different feelings of the sprite (see Figure 7).



**Figure 6.** The code for creating emotions using machine learning

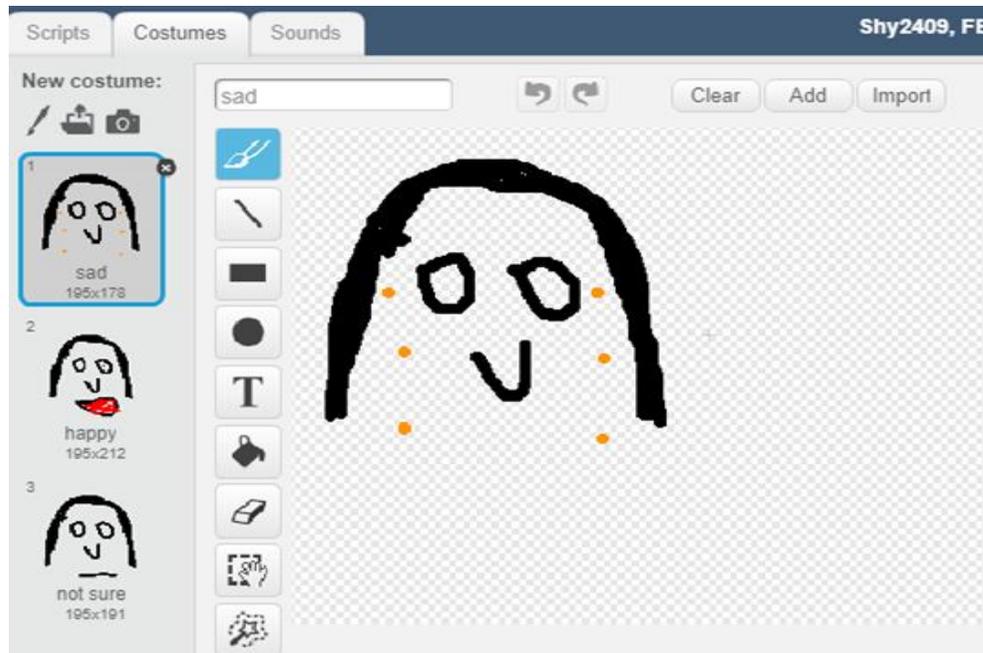


Figure 7. The sprites in Scratch to create emotions

### The Computational STEAM Pedagogy

According to Jona et al. (2014) “by pairing computational thinking instruction with STEM content, students can explore and apply computational approaches within more established and accessible STEM context.” Additionally, by spreading Computational Thinking skills across the STEM spectrum, students will be exposed to these ideas on different occasions across multiple years and across different content areas. In this way, STEM can enrich computational learning. Research has also shown that the reverse is true; the use of computational tools has been shown to enable deeper learning of STEM content areas for students” (National Research Council, 2011a,b; Repenning, Webb, & Ioannidou, 2010; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013). In conclusion, “Computation is an indispensable component of STEM disciplines as they are practiced in the professional world. In the last twenty years, nearly every STEM field has seen the birth or reconceptualization of a computational Counterpart” (Jona et al., 2014).

The concepts of computing and computational are discussed explicitly in (Yaşar et al., 2016). Authors state that Computational Pedagogy is an inherent outcome of Computing, Mathematics, Science and Technology integration. In the same article, computing is related to algorithmic and programming. They also suggest that computational modelling and simulation technology (CMST) can be used to improve technological pedagogical content knowledge (TPACK) of

teachers. According to cognitive psychology, exposure to new concepts through links to multiple views from different fields of study is an effective retrieval strategy recommended by cognitive psychologists. This is called interleaved retrieval practice and it forms a cognitive foundation for the interdisciplinary computational pedagogical content knowledge (CPACK) framework that has been developed recently by computational science practitioners and educators (Yaşar et al., 2015). This multi-faceted interdisciplinary knowledge domain has been called Computational Pedagogical Content Knowledge (CPACK) domain framework. When Mathematics, Computing, and Sciences are integrated, this can give birth not only to a new content domain of Computational Science, as witnessed by degree programs in the past two decades but also a particular Computational Pedagogy.

In our model we integrate the inquiry based teaching and learning approach, the Computational experiment spaces, the Engineering Education Epistemology (EEE) and STEM (see Figure 8) content transdisciplinary approach and we call our model of teaching and learning—Computational STEAM Pedagogy (CSAP) (Table 3), with easy extension to include A(Arts) (Psycharis, 2018). In our model, we have integrated the Computational Experiment for Education, the CPACK, the STEM content inter-disciplinary approach, the Engineering Education Epistemology, Art epistemology and the features of inquiry based teaching and learning approach.

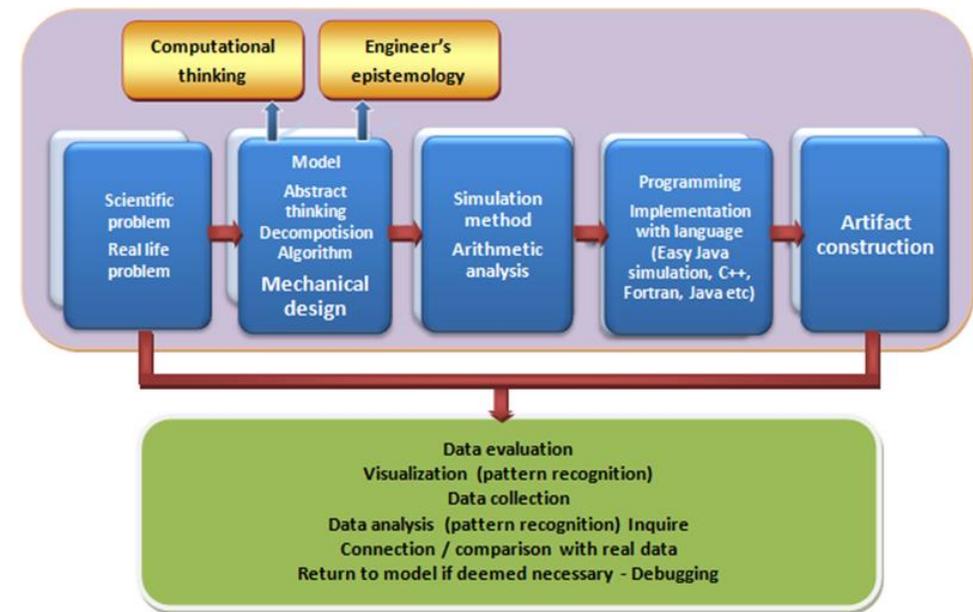


Figure 8. The Computational STEAM model.

**Table 3.** The Computational STEAM content Pedagogy

<b>Spaces of the Computational Experiment</b>	<b>Essential Features of Inquiry</b>	<b>Spaces of the Computational Experiment</b>
<b>Hypotheses space</b>	✓ <u>Essential Features of Inquiry</u>	Orienting and asking questions; generating hypotheses
	✓ Question	
	✓ <u>Dimensions of CT</u>	
	✓ Abstraction, decomposition	
	✓ <u>STEM Epistemology</u>	
<b>Experimental space</b>	✓ <u>Use of a product from real life</u> - Unplugged activities	Planning Investigating Analysis and interpretation Modelling
	✓ <u>EEE</u>	
	✓ Provision of Engineering products in a form of a video, picture, artifact	
	✓ Art design and plan	
	✓ <u>Essential Features of Inquiry</u>	
<b>Prediction Space</b>	✓ Evidence, Analyze, Explain	Conclusion-Evaluation-Prediction
	✓ <u>Dimensions of CT</u>	
	✓ Abstraction, algorithmic thinking	
	✓ <u>STEM Epistemology</u>	
	✓ Intertwine science and mathematics to model the phenomenon	
<b>Prediction Space</b>	✓ Creation of Code to control artefacts –maybe use of physical computing	Conclusion-Evaluation-Prediction
	✓ <u>EEE</u>	
	✓ Design of artefact based on the simulation-revision –if necessary- of the prototype	
	✓ Art integration, possibly through code	
	✓ Art integration trough “unplugged artifact”	
<b>Prediction Space</b>	✓ <u>Essential Features of Inquiry</u>	Conclusion-Evaluation-Prediction
	✓ Connect, Communicate	
	✓ <u>Dimensions of CT</u>	
	✓ Debugging and generalization	
	✓ <u>STEM Epistemology</u>	
<b>Prediction Space</b>	✓ Generalize the methodology to other similar cases, maybe use of remixing	Conclusion-Evaluation-Prediction
	✓ <u>EEE</u>	
	✓ Provide design patterns for future use-metacognitive experiences	

## References

- Aho, A. V. (2012). Computation and computational thinking. *Computer Journal*, 55(7), 832 – 835.
- Antola Crowe, H., Brandes, K., Davison Avilés, B., Erickson, D., & Hall, D. (2013). Transdisciplinary teaching: Professionalism across cultures. *International Journal of Humanities and Social Science*, 3(13), 194–205.
- Assay, L. D., & Orgill, M. K. (2010). Analysis of Essential Features of Inquiry Found in Articles Published in The Science Teacher, 1998–2007. *Journal of Science Teacher Education*, 21(1), 57-79, DOI: 10.1007/s10972-009-9152-9
- Ates, S., & Cataloglu, E. (2007). The effects of students’ reasoning abilities on conceptual understandings and problem-solving skills in introductory mechanics. *European Journal of Physics*, 28(6), 1161–1171.
- Barr, V., & Stephenson, C. (2011). Bringing Computational Thinking to K-12: What Is Involved and What Is the Role of the Computer Science Education Community? *ACM Inroads*, 2(1), 48-54. doi:10.1145/1929887.1929905
- Bell, P. L., Hoadley, C., & Linn, M. C. (2004). Design-based research as educational inquiry. In M. C. Linn, E. A. Davis & P. L. Bell (Eds.), *Internet environments for science education*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Bell, T., Urhahne, D., Schanze, S., & Ploetzner, R. (2010). Collaborative inquiry learning: models, tools and challenges. *International Journal of Science Education*, 32(3), 349-377.
- Bequette, M., & Bequette, J. (2011). *STEM plus arts make STEAM? Effective integration of aesthetic-based problem solving across topic areas*. STEM Colloquium. Minnesota.
- Beth, E. W., & Piaget, J. (1966). *Mathematical Epistemology and Psychology*. Reidel: Dordrecht, The Netherlands.
- Bienkowski, M., Snow, E., Rutstein, D. W., & Grover, S. (2015). *Assessment design patterns for computation-al thinking practices in secondary computer science: A first look (SRI technical report)*. Menlo Park, CA: SRI International.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K., (2016). Developing computational thinking in compulsory education - Implications for policy and practice. *JRC Science for Policy Report*. doi:10.2791/792158.
- Borrego, M., & Newswander, L. K. (2008). Characteristics of Successful Cross-disciplinary Engineering Education Collaborations. *Journal of Engineering Education*, 97(2), 123-134.
- Breiner, J. M., Harkness, S. S., Johnson, C. C., & Koehler, C. M. (2012). What is STEM? A discussion about conceptions of STEM in education and partnerships. *School Science and Mathematics*, 112(1), 3–11.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*, Vancouver, BC, Canada, 13–17 April 2012.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Bybee, R. W., Trowbridge, L. W. & Powell, J. C. (2008). *Teaching Secondary School Science: Strategies for Developing Scientific Literacy*. New Jersey Merrill.
- Campbell, D. & Samsel, F. (2015) Pursuing value in art-science collaborations. *IEEE Comput. Graph. Appl.*, 35(1):6-11. DOI: 10.1109/MCG.2015.18.
- Carlson, L. E., & Sullivan, J. F. (2004). Exploiting design to inspire interest in engineering across the K-16 engineering curriculum. *International Journal of Engineering Education*, 20(3), 372-378.
- Cetin, I., & Dubinsky, E. (2017). Reflective abstraction in computational thinking. *The Journal of Mathematical Behavior*, 47, 70-80.
- Chande, S. (2015). A Conceptual Framework for Computational Thinking as a Pedagogical Device. *International Journal of Innovative Research in Computer and Communication Engineering*, 3(11), 11682-11688.
- Chandler, J., Fontenot, A. D., & Tate, D. (2011). Problems associated with a lack of cohesive policy in K-12 precollege engineering. *Journal of Pre-College Engineering Education Research (J-PEER)* 1, 5.
- Chittleborough, G., & Treagust, D. (2007). The modelling ability of non-major chemistry students and their understanding of the sub-microscopic level. *Chemistry Education Research and Practice*, 8 (3), 274-292.
- Computing at School Working Group (2012). *Computer Science: A Curriculum for Schools*. Available at <https://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>

- Cross, J. (2017). Doctoral dissertation). *Technical Report CMU-RI-TR-17-30*. Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.
- Daugherty, M. (2013). The Prospect of an 'A' in STEM Education. *The Journal of STEM Education*, April-June 2013, 10-15.
- Dede, C., Mishra, P., & Voogt, J. (2013). *Working group 6: Advancing computational thinking in 21st century learning*. [http://www.curtin.edu.au/edusummit/local/docs/Advancing\\_computational\\_thinking\\_in\\_21st\\_century\\_learning.pdf](http://www.curtin.edu.au/edusummit/local/docs/Advancing_computational_thinking_in_21st_century_learning.pdf) Accessed 28 January 2017.
- Denning, P. J. (2003). Great Principles of Computing. *Communications of the ACM*, 46(11), 15-20.
- Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM*, 50, 13-18.
- Egodawatte, G. (2011). *Secondary School students' misconceptions in Algebra*. Unpublished Doctoral Dissertation. University of Toronto, Toronto, CA.
- Eisner, E. W. (2002). *The arts and the creation of mind*. Yale University Press.
- Feldhausen, R., Weese, J. L., & Bean, N. H. (2018). Increasing Student Self-Efficacy in Computational Thinking via STEM Outreach Programs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, February (pp. 302–307).
- Ghanbari, S. (2015). Learning across disciplines: A collective case study of two university programs that integrate the arts with STEM. *International Journal of Education & the Arts*, 16(7). Retrieved from <http://www.ijea.org/v16n7/>
- Goris, T. V., & Dyrenfurth, M. J. (2010) Students' Misconceptions in Science, Technology and Engineering. *ASEE Illinois/Indiana Section Conference Proceeding*, Purdue University, West Lafayette, IN. Retrieved from <http://ilin.asee.org/Conference2010/Papers2010.html>
- Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, 51(8), 25-27. <https://doi.org/10.1145/1378704.1378713>
- Hestenes, D. (1999). The scientific method. *American Journal of Physics*, 67, 273-276.
- Hußmann, S., Schacht, F., & Schindler, M. (2019). Tracing conceptual development in mathematics: epistemology of webs of reasons. *Mathematics Education Research Journal*, 31(2), 133-149.
- International Technology Education Association (ITEA) (2007). *Standards for Technological Literacy: Content for the Study of Technology*. ITEA: Reston, VA.
- Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In *Future Directions in Computer Science Education Summit Meeting*, Orlando, FL.
- Justi, R. S. & Gilbert, J. K., (2002). Models and modelling in chemical education, In J. Gilbert, O. De Jong, R. Justi, D. F. Treagust and J. H. Van Driel (Eds.), *Chemical education: towards research-based practice* (pp. 213 - 234), Dordrecht: Kluwer Academic Publishers.
- Juszczak, M. D. (2015). From Towards a Computational Pedagogy – Analysis of ABM Deployment in Pedagogical Instances. *International Journal of Pedagogy Innovation and New Technologies*, 2(1), 2-13 DOI: 10.5604/23920092.1159113
- Katchi, L., Pearson G., & Feder, M. (2009). *Engineering in K-12 education: Understanding the status and improving the prospects*. Washington, DC: National Academy of Engineering and National Research Council.
- Klahr, D., & Dunbar, K. (1998). Dual space search during scientific reasoning. *Cognitive Science*, 12, 1-48.
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A pedagogical framework for computational thinking. *Digital Experiences in Mathematics Education*, 3(2), 154-171.
- Kroes, P., & Van de Poel, I. (2009). Problematising the notion of Social Context of Technology. In S. H. Christensen, B. Delahousse, & M. Meganck (Eds.), *Engineering in Context* (pp. 61-74). Denmark: Academica.
- Landau, R. H., Páez, J. & Bordeianu, C. (2008). *A Survey of Computational Physics: Introductory Computational Science*. Princeton and Oxford: Princeton University Press.
- Lawson, A. E., Banks, D. L., & Logvin, M. (2007). Self-efficacy, reasoning ability, and achievement in college biology. *Journal of Research in Science Teaching*, 44(5), 706–724.
- Libow Martinez, S. & Stager, G. (2013). *Invent to Learn - Making, Tinkering, and Engineering in the Classroom*. Torrance, CA: Constructing Modern Knowledge Press.
- Magnani, L., & Nersessian, N. (eds.) (2002). *Model-Based Reasoning: Science, Technology, Values*. Dordrecht: Kluwer
- Martin, F., Greher, G., Heines, J, Jeffers, Kim, H.J., Kuhn, S., Roehr, K., Selleck, N., Silka, L., & Yanco, H. (2009): Joining computing and the arts at a mid-size university. *J. Comput. Sci. Coll.* 24(6), 87–94.
- McGregor, S. (2015). Transdisciplinary knowledge creation. In P. T. Gibbs (Ed.), *Transdisciplinary professional learning and practice* (pp. 9-24). New York, NY: Springer.
- Mendonça, P.C.C. & Justi, R. (2013). An instrument for analyzing arguments produced in modeling-based chemistry lessons. *Journal of Research in Science Teaching*, 51(2), 192–218. doi:10.1002/tea.21133.
- Mishra, P. & Yadav, A. (2013). Of art and algorithms: Rethinking technology & creativity in the 21st century. *TechTrends*, 57(3), 11.
- Mitcham, C. (1994). *Thinking through technology: the path between engineering and philosophy*. Chicago: the University of Chicago Press.
- Moore, T. J. (2008). *STEM integration: Crossing disciplinary borders to promote learning and engagement*. Invited presentation to the faculty and graduate students of the UTeachEngineering, UTeachNatural Sciences, and STEM Education program area at University of Texas at Austin, December 15, 2008.
- Moreno-León, J., Robles, G., & Román-González, M.D. (2015). Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 46, 1–23.
- National Academy of Engineering (NAE) & National Research Council (NRC). (2014). *STEM integration in K-12 education: status, prospects, and an agenda for research*. The National Academies Press, Washington, DC.
- National Academies of Sciences, Engineering and Medicine, (2014). *STEM Integration in K-12 Education: Status, Prospects, and an Agenda for Research*. Retrieved from <http://nap.edu/18612>
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. The National Academies Press, Washington, DC.
- National Research Council (2011a) *Learning science through computer games and simulations*. The National Academies Press, Washington, DC.
- National Research Council. (2011b) *Report of a workshop of pedagogical aspects of computational thinking*. The National Academies Press, Washington, DC.
- NGSS Lead States (2013). *Next generation science standards: for states, by states*. The National Academies Press, Washington, DC.
- Nersessian, N. (1992). How do scientists think? Capturing the dynamics of conceptual change in science. In R. Giere (Ed.), *Cognitive models of science* (pp. 3-44). Minneapolis: University of Minnesota Press.
- Nicolescu, B. (2002). *Manifesto of transdisciplinarity* [Trans. K-C. Voss]. New York, NY: SUNY.
- Nicolescu, B. (2004). Gurdjieff's philosophy of nature. In J. Needleman & G. Baker (Eds.), *Gurdjieff* (pp. 37-69). New York, NY: The Continuum International Publishing Group
- Nievergelt, J. (1974). Computers and Mathematics Education. *Compo & Moths. with Appls*, 1, 121-132, Pergamon Press.
- Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, New York, 1992
- OpenLearn from The Open University. (n.d. a). Retrieved from <https://www.open.edu/openlearn/science-maths-technology/computing-and-ict/introduction-computational-thinking/content-section-2>
- OpenLearn from The Open University. (n.d. b). Retrieved from <https://www.open.edu/openlearn/science-maths-technology/computing-and-ict/introduction-computational-thinking/content-section-2.5>
- Pedaste, M., & Palts, T. (2017). Tasks for Assessing Skills of Computational Thinking. *The 2017 ACM Conference*.
- Piaget, J. (1970). *Genetic epistemology*. New York: Columbia University Press.

- Przybylla, M., & Romeike, R. (2014). Physical computing and its scope - towards a constructionist computer science curriculum with physical computing. *Informatics in Education*, 13(2), 225-240.
- Psycharis, S. (2013). The Effects of the Computational Models on Learning Performance, Scientific Reasoning, Epistemic Beliefs and Argumentation. *Computers & Education*, 68, 253–265. DOI: 10.1016/j.compedu.2013.05.015
- Psycharis, S. (2016a). The Impact of Computational Experiment and Formative Assessment in Inquiry Based Teaching and Learning Approach in STEM Education. *Journal of Science Education, and Technology*, 25(2), 316-326 DOI: 10.1007/s10956-015-9595-z
- Psycharis, S., (2016b). Inquiry Based- Computational Experiment, Acquisition of Threshold Concepts and Argumentation in Science and Mathematics Education. *Educational Technology & Society*, 19(3), 282–293.
- Psycharis, S., Kalovrektis, K., Sakelalridi, E., & Korres, K., Mastorodimos, D. (2018). Unfolding the Curriculum: Physical Computing, Computational Thinking and Computational Experiment in STEM's Transdisciplinary Approach. *European Journal of Engineering Research and Science. Special Issue: CIE 2017*, 19-27. <https://doi.org/10.24018/ejers.2018.0.CIE.639>
- Psycharis, S. (2018). STEAM in Education: A Literature review on the role of Computational Thinking, Engineering Epistemology and Computational Science. Computational STEAM Pedagogy (CSP). *Scientific Culture*, 4(2), 51-72. <https://sci-cult.com>
- Psycharis, S., & Kotzampasaki, E. (2019). The Impact of a STEM Inquiry Game Learning Scenario on Computational Thinking and Computer Self-confidence. *Eurasia Journal of Mathematics, Science and Technology Education*, 15(4), em1689. <https://doi.org/10.29333/ejmste/103071>
- Psycharis, S. (2019). *Relogia Interview*, Art-sci-technology- relogia.net. Sofia.
- Qiu, K., Buechley, L., Baafi, E. & Dubow, W. (2013). A curriculum for teaching computer science through computational textiles. *Proc. 12th Int. Conf. Interact. Des. Child. - IDC '13*, pp. 20–27.
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 265–269).
- Rubio, M.A., Manoso-Hierro, C. & Perez de Madrid y Pablo, A. (2013). Using Arduino to enhance computer programming courses in science and engineering. *Proceedings of EDULEARN13 Conference 1st-3rd July 2013, Barcelona, Spain*.
- Schnittka, C. G., Bell, R. L., & Richards, L. G. (2010). Save the penguins: Teaching the science of heat transfer through engineering design. *Science Scope*, 34(3), 82-91.
- Schulz, S. & Pinkwart, N. (2015). *Physical Computing in STEM Education*. Retrieved from: <https://cses.informatik.hu-berlin.de/pubs/2015/wipsce/physical-computing-in-stem-education.pdf>
- Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies*, 18(2), 351-380.
- Sentance, S., & Csizmadia, A. (2015). *Teachers' perspectives on successful strategies for teaching Computing in school*. Paper presented at IFIP TCS, 2015. <http://community.computingatschool.org.uk/files/6769/original.pdf>
- Shunn, C., & Klahr, D. (1995). *A 4-space model for scientific discovery*. In Paper presented at the AAAI symposium systematic methods of scientific discovery, Menlo Park, CA.
- Sloot, P. (August 1994). *Lecture on parallel scientific computing and simulations*. Sopron, Hungary: CERN School on Computing.
- Swoyer, C. (1991). Structural Representation and Surrogate Reasoning. *Synthese*, 87, 449-508.
- Toomey, A.H., Markusson, N., Adams, E. & Brockett, B. (2015). *Inter- and trans-disciplinary research: a critical perspective. G.SDR 2015 Brief*. Retrieved from <https://sustainabledevelopment.un.org/content/documents/612558-Inter-%20and%20Trans-disciplinary%20Research%20-%20A%20Critical%20Perspective.pdf>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715-728. <https://doi.org/10.1007/s10639-015-9412-6>
- Weese, J., & Feldhausen, R. (2017). *STEM Outreach: Assessing Computational Thinking and Problem Solving*. 2017 ASEE Annual Conference & Exposition. ASEE Conferences, Columbus, Ohio. <https://peer.asee.org/28845>
- Weese, J. L., Feldhausen, R., & Bean, N. H. (2016). The Impact of STEM Experiences on Student Self-Efficacy in Computational Thinking. *Proceedings of the 123rd American Society for Engineering Education Annual Conference and Exposition* (ASEE 2016). New Orleans, LA, USA.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining Computational Thinking for Mathematics and Science Classrooms. *Journal of Science Education and Technology*, 25(1), 127-147. DOI: 10.1007/s10956-015-9581-5
- White, B., Frederiksen, J., & Spoehr, K. (1993). Conceptual models for understanding the behavior of electrical circuits. In M. Caillot (Ed.) *learning Electricity and Electronics with Advanced Educational Technology* (pp. 77-95). New York: Springer-Verlag.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
- Wolfram, S. (2002). *A New Kind of Science* (1st ed.) Wolfram Media: Tokyo.
- Xie, C., Tinker, R., Tinker, B., Pallant, A., Damelin, D., & Berenfeld, B. (2011). Computational experiments for science education. *Science*, 332, 1516–1517.
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S. & Korb, J.T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education, ACM*, pp 465–470.
- Yaşar, O. & Landau, R. (2003): Elements of CSE Education. *SLAM Review*, 45(4), 787–805.
- Yaşar, O. (2004). Computational math, science and technology: A new pedagogical approach to math and science education. In Lagana, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) *ICCSA 2004*. LNCS, 3045, 807-816. Springer: Heidelberg.
- Yaşar, O. (2013). Teaching Science through Computation. *International Journal of Science, Technology and Society*, 1(1), 9-18. DOI: 10.11648/j.ijsts.20130101.12
- Yaşar, O., Veronesi, P., Maliekal, J. and Little, L. (2015). Computational Pedagogical Content Knowledge (CPACK). In D. Slykhuus & G. Marks (Eds.), *Proceedings of Society for Information Technology & Teacher Education Conference 2015* (pp. 3514-3521).
- Yasar, O., Veronesi, P., Maliekal, J., Little, L. J., Vattana, S. E., & Yeter I. H. (2016). *Computational Pedagogy: Fostering a New Method of Teaching*. Presented at: ASEE Annual Conference and Exposition. Presented: June 2016. Project: SCOLLARCIT.
- Zendler, A. & Spannagel, C. (2008). Empirical Foundation of Central Concepts for Computer Science Education. *ACM Journal on Educational Resources in Computing*, 8(2), 6.
- Zhang, Y. & Luo, C. (2012). *Training for Computational Thinking Capability on Programming Language Teaching*. The 7th International Conference on Computer Science & Education (ICCSE 2012), 1804-1809. IEEE. DOI: 10.1109/ICCSE.2012.6295420

