Conditions and Problems of Computational Thinking Instruction in Lower Secondary Schools

สภาพและปัญหาการจัดการเรียนการสอนที่ส่งเสริมทักษะการคิดเชิงคำนวณ ในโรงเรียนระดับชั้นมัธยมศึกษาตอนต้น

Kitiya Middleton[1] and Praweenya Suwannatthachote[2]
กิติยา มิดเดิลตัน[1] และ ปราวีณยา สุวรรณณัฐโชติ[2]

Article History
Receive: August 10, 2021
Revised: December 13, 2021
Accepted: December 14, 2021

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อศึกษาสภาพปัญหาการจัดเรียนการสอนในการจัดการเรียนการสอนที่ส่งเสริมทักษะการคิดเชิงคำนวณ และเพื่อเสนอแนวทางการพัฒนารูปแบบการจัดการเรียนการสอนที่ส่งเสริมทักษะการคิดเชิงคำนวณ วิธีดำเนินการวิจัยเป็นการวิจัยเชิงปริมาณและเชิงคุณภาพ โดยเก็บรวบรวมข้อมูลจากครูผู้สอนรายวิชาเกี่ยวกับเทคโนโลยี จำนวน 24 คน จากการใช้แบบสอบถามและแบบสัมภาษณ์ที่ผ่านการตรวจสอบคุณภาพของเครื่องมือจากผู้ทรงคุณวุฒิ จำนวน 5 ท่าน (ดัชนีความสอดคล้องระหว่างข้อคำถามและจุดประสงค์ (Item-Objective Congruence Index : IOC) = 0.98) วิเคราะห์ข้อมูลโดยการหาค่าความถี่ ค่าร้อยละ ค่าเฉลี่ย และค่าเบี่ยงเบนมาตรฐาน ในแต่ละประเด็นของข้อคำถาม รวมทั้งวิเคราะห์แบบอุปนัย โดยวิธีตีความ และสร้างข้อสรุปจากข้อมูลที่ได้จากการสัมภาษณ์ ผลการวิจัยพบว่า ครูผู้สอนมีความคิดเห็นว่านักเรียนมีปัญหาการคิดเชิงคำนวณในด้านการแจกแจงปัญหาเป็นส่วนย่อย โดยปัจจัยที่มีผลต่อการสร้างความเข้าใจในการคิดเชิงคำนวณของนักเรียน ได้แก่ 1) นักเรียน 2) ครูผู้สอน 3) กิจกรรมการเรียนรู้ และ 4) อื่นๆ ซึ่งมีวิธีการในการเตรียมความพร้อมนักเรียนโดย 1) การวิเคราะห์เนื้อหา/ผู้เรียน/จุดประสงค์ 2) การเตรียมสื่อและอุปกรณ์ และ 3) การออกแบบกิจกรรมการเรียนรู้ จากผลการวิจัยดังกล่าวสามารถเสนอแนวทางการพัฒนาการจัดการเรียนการสอน ประกอบด้วย การวิเคราะห์เนื้อหา/ผู้เรียน/จุดประสงค์ เตรียมสื่อและอุปกรณ์ และการออกแบบกิจกรรมการเรียนรู้ เลือกวิธีการสอนที่หลากหลายและเหมาะสมกับผู้เรียน เสริมสร้างประสบการณ์ในการคิดเชิงคำนวณทั้งในและนอกห้องเรียน สร้างเครือข่ายระหว่างครูและบุคลากรทางการศึกษา นักเรียน และส่งเสริมการวิจัยทางการคิดเชิงคำนวณ ประโยชน์ที่ได้รับจากการวิจัย คือ เป็นข้อมูลให้ครูผู้สอนพัฒนาและปรับปรุงวิธีการและรูปแบบการจัดการเรียนการสอนวิทยาการคำนวณ

คำสำคัญ : สภาพและปัญหา ; ทักษะการคิดเชิงคำนวณ ; การเรียนการสอน ; โรงเรียนระดับชั้นมัธยมศึกษาตอนต้น ; แนวทางการพัฒนารูปแบบการจัดการเรียนการสอน

---

[1] นักศึกษาปริญญาเอก สาขาเทคโนโลยีและสื่อสารการศึกษา คณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, Doctoral Student in Department of Educational Technology and Communications, Faculty of Education, Chulalongkorn University
[2] ผู้ช่วยศาสตราจารย์ คณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, Associate Professor, Faculty of Education, Chulalongkorn University

ABSTRACT

The objectives of this study were to examine the conditions and problems of computational thinking instruction in lower secondary students and to suggest directions for developing a learning model that supports computational thinking in lower secondary students. This research project applied quantitative and qualitative research methods by collecting data from twenty-four teachers. The methodologies were also included survey and an interview with five experts. (Item Objective Congruence: IOC = 0.98). The data were analyzed by finding frequency, percentage, mean ($\overline{X}$) and standard deviation (S.D.) in each question. We also included inductive analysis by interpretation and drew conclusions from the information obtained from the interviews. The study revealed that teachers agreed that there are problems in computational thinking in the areas of problem decomposition and modularity. There are four factors affecting students' comprehension of computational thinking: 1) students; 2) teachers; 3) learning activities; and 4) other factors. Methods used to prepare students were first, analyzing the content / students / purposes; second, preparation of media and equipment, and third, the design of learning activities. The research findings, suggested that directions for developing a learning model consist of analyzing contents, students, and purpose, preparation of instructional media, and design of learning activities. Teachers should choose the teaching method that is most suitable for the students. Create computational thinking experience for students both in the classroom and outside the classroom; and create a community about computational thinking. Furthermore, teachers should develop their own computational thinking skills continually and should also promote computational thinking research. The research's contribution is providing information for teachers to develop and improve teaching methods and models of computational thinking.

Keywords : Conditions and Problems ; Computational Thinking ; Instruction ; Lower Secondary Schools ; Directions to Develop a Learning Model

Introduction

Technology has been rapidly changing life, careers, and communication. Educators around the world need to adjust their education systems to keep up with this changing technology and prepare students for the increasing importance of global economics. (International Society for Technology in Education, 2017) Nowadays, the way computer technology is applied and the way it works have been changed in almost all disciplines.

Computers should be viewed as tools. Tools to help us and not hinder us; to benefit our lives and not complicate our lives. We must understand how computers work and what benefits computers bring. Computational thinking infuses important lessons and principles from computer science, the subject that teaches us how to use computers to compliment our will to learn and create. These lessons and principles include how to choose the prominent details of a problem, how to develop a problem-solving process in ways that the process can be automated. These processes should be important to everyone, regardless of whether they work in the computer field or not, because the ultimate goal of refining them is to be able to not only solve a problem, but also to integrate a computer in a solution to execute the task more quickly and efficiently. (Karl, 2017)

Nowadays, the focus is on Computational Thinking (CT) as a part of twenty-first century skills for every student, which leads to creating curriculum as formal education for the classroom that will challenge and support young students by sharpening their critical thinking skills (Yadav, Good, Voogt, and Fisser, 2017)

Furthermore, the International Society for Technology in Education (ISTE) has established standards for students to provide clear guidance for the development of essential skills and knowledge: 1) empowered learner 2) digital citizen 3) knowledge constructor 4) innovative designer) 5) computational thinker 6) creative communicator and 7) global collaborator. From these standards, it can be shown that the ISTE have seen the importance of CT as one of the standards by which students create and utilize strategies for comprehending and solving problems in ways that harness the power of technological methods to develop and test solutions. (International Society for Technology in Education, 2017)

Recently, there has been a growing interest in CT. It is indicative that a large number of studies focusing on CT have been published in recent years (Tikva and Tambouris, 2021). This large body of literature indicates challenges in particular areas including: (a) exploring and reviewing the studies on the conception, domain of CT (Hickmott, Prieto-Rodriguez, and Holmes, 2018; Tikva and Tambouris, 2021; Yadav et al., 2017) and (b) exploring teaching methods, approaches, and pedagogical strategies with CT (Caeli and Yadav, 2020; Flórez et al., 2017; Kale et al., 2018; Lyon and Magana, 2021) and (c) developing frameworks, tools, and assessments (Asbell-Clarke et al., 2021; Brasiel et al., 2017; Lawanto, Close, Ames, and Brasiel, 2017; Repenning, Basawapatna, and Escherle, 2017; Yildiz Durak, 2020).

From that context Thailand has focused on twenty-first century skills by establishing an educational policy in the National Education Act B.E. 2542 and Amendments (Second National Education Act B.E. 2545 (2002) and Third National Education Act B.E. 2553 (2010) National Education Guidelines Section 24 specified that in organizing the learning process, educational institutions and agencies concerned shall provide training in thinking process, management, and how to face various situations and application of knowledge for obviating and solving problems. This is consistent with the National Education Plan (2017-2036) that aims to equip students with basic skills and characteristics of Thai citizens and 21st century skills and attributes, and to promote the good habit of encouraging people of all ages to have the skills, knowledge and abilities to improve the quality of life appropriately according to the potential of each age. In addition, the National Education Plan set the role of the learner to learn and practice thinking skills and to improve reasoning and group processing skills (Office of the Education Council, 2017). The Ministry of Education, the governing office that manages education in Thailand, conducted a review of the Core Curriculum of Basic Education of 2008 and revised this document to be in line with the economic, societal, cultural, and environmental changes that took place to be consistent with science and technology that progressed rapidly. The revised curriculum is The Basic Education Core Curriculum (2017). By using data from the 12^{th} National Economic and Social Development Plan, 20-Year National Strategy and the National Education Plan 2017-2036 was implemented as a framework and direction for curriculum development to be more appropriate. One of the main goals was to improve the curriculum regarding mathematics, science, and social studies, which are important to the development of the country. These subjects are the cornerstone of human creativity, rational thinking, as well as systematic and thorough analysis of problems or situations. Furthermore, these subjects can be used in daily life as well as to utilize technology in the integration of scientific and mathematical knowledge to solve problems or develop work using engineering design processes leading to the inventions or innovations that benefit life. These subjects, in addition to using computational thinking skills, help create knowledge of computer science and technology and communication to solve problems encountered in real life effectively. (Office of the Basic Education Commission, 2017)

In conclusion, because of the problems and findings mentioned above, researchers are interested in examining the conditions and problems of computational thinking instruction in lower secondary students to find the guidelines that will lead to the best practice of the most effective computational thinking education.

Objective

1. To examine the conditions and problems of computational thinking instruction in lower secondary students.

2. To suggest directions to develop a learning model that encourages computational thinking in lower secondary school students.

Literature Review

This study aims to examine the conditions and problems of computational thinking instruction in lower secondary students and to suggest directions to develop a learning model that encourages computational thinking in lower secondary students. To provide better understandings of the research questions, the following literature reviews include computational thinking, teaching strategies, and scaffolding.

1. Computational Thinking

CT was first mentioned in a study by Seymour Papert who explains a method of developing thinking processes through programming. (Rose, Habgood and Jay, 2017) Recently CT has been analyzed to understand and describe various ways CT can be applied to the educational process. CT can be used to guide and structure a course as well as measure students' performance (Lyon and Magana, 2021).

Computational thinking can be defined as the thought processes (Gleasman and Kim, 2020; Lee, Grover, Martin, Pillai, and Malyn-Smith, 2020) that involves solving problems, designing systems, and understanding the behavior of humans (Korkmaz, Çakir and Özden, 2017; Yildiz Durak, 2020) to formulate problems and the solutions (Lee et al., 2020; Malyn-Smith and Angeli, 2020) while having the knowledge, skills, and attitudes necessary to be able to use the computers to solve the problems (Korkmaz et al., 2017; Song, Hong and Oh, 2021; Yildiz Durak, 2020).

There have been studies that explore and review on the concept and domain of CT (Hickmott et al., 2018; Tikva and Tambouris, 2021; Yadav et al., 2017) and investigate the teaching methods, approaches, and pedagogical strategies with CT (Caeli and Yadav, 2020; Flórez et al., 2017; Kale et al., 2018; Lyon and Magana, 2021) and developing frameworks, tools and assessments (Asbell-Clarke et al., 2021 ; Brasiel et al., 2017 ; Lawanto et al., 2017 ; Repenning et al., 2017 ; Yildiz Durak, 2020).

2. Teaching Strategies

Many studies have reported that most of the instruction in computational thinking occurs through problem-solving transfer with programming explained (Kale et al., 2018). The findings from Hsu, Chang, and Hung (2018) showed that there are sixteen learning strategies that have been used in the past studies, including problem-based learning, collaborative learning, project-based learning, game-based learning, scaffolding, storytelling, computational learning theory, aesthetic experience, concept-based learning, embodiment-based learning, human-computer interaction teaching, and universal design for learning. Moreover, Kale et al (2018) argued that there are three strategies that can help teachers make the connections between computational thinking and their teaching. The first strategy emphasized content specific examples regarding the use of computational thinking tools. The second strategy was based on the problem-solving nature of computational thinking. The third strategy built on the first two and it was the main theme of the paper. It focused on the methods of teaching problem-solving as a means to support the kind of knowledge necessary to the processes involved in computational thinking.

In Thailand, there have been studies that develop learning achievement of students on CT via project-based learning (Khamnaen, 2019) problem-based learning (Daungjun, 2018) peer assisted learning (Supaluk, 2018) and gamification (Daungjun, 2018; Amnuayporn, Yampinij, Meejaleurn and Supaluk, 2019).

3. Scaffolding

Researchers value analyzing learners' coding patterns and the connection of the learners with their performance in order to find the best way to provide scaffolding for the learner. Professionals in the educational field seek a way to create a plausible solution to meet the specific instructional goals when teaching computational skills.

Scaffolding in its simplest term is a process that guides and supports the learner who are unable or require assistance to complete the task (Eggen and Kauchak, 2011; Hannafin, Land, and Oliver, 1999 ; Saputri and Wilujeng, 2017 ; Vygotsky, 1979). Scaffolding is designed to only be temporary (Vygotsky, 1979). Once the scaffolding has been omitted in learning, it is believed that the learner will prevail at their task with consistent improvement (Eggen and Kauchak, 2016 ; Vygotsky, 1979).

Scaffolding has been used immensely in computational thinking. Saputri and Wilujeng (2017) stated that scaffolding is extremely beneficial for learners and their ability to solve problems. Basu, Biswas, and Kinnebrew (2017) deduced that the effects of scaffolding stretch far beyond the learners' performances and their strategy to use their abilities to complete their modeling tasks and to comprehend and associate the representations at different levels.

In Thailand, scaffolding has been used immensely in many disciplines that involve problem solving. For example, Teemueangsa and Jedaman (2021) use scaffolding as one of their learning strategies on developing science learning activities using online gamification. Threekunprapa and Yasri (2020) proposed a 3S self-directed learning approach for fostering the CT development, composing of self-check (in pairs), self-debug (in pairs), and scaffolding. Haruehansawasin and Kiattikomol (2018) investigated scaffolding approaches for supporting low-achieving learners in a problem-based learning environment.

Conceptual Framework

In this study, the researchers analyzed and synthesized ideas, theories and principles related to computational thinking (Gleasman and Kim, 2020; Korkmaz et al., 2017; Lee et al., 2020; Malyn-Smith and Angeli, 2020; Song et al., 2021; Yildiz Durak, 2020), teaching strategies (Buitrago Flórez et al., 2017; Hsu et al., 2018; Kale et al., 2018), and scaffolding which can be summarized as a conceptual framework as show in Figure 1.

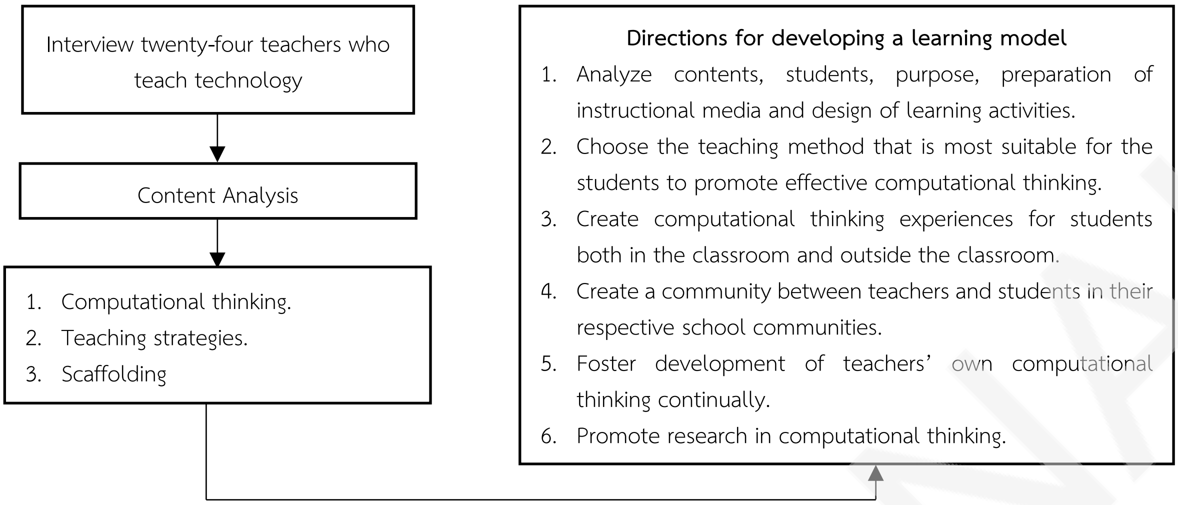

Figure 1 Conceptual Framework

Research Methodology

Population and Samples

The participants in this study were 8,853 technology teachers classified by the four regions, namely the North, Northeast, Central, and the South of Thailand, as well as classified by school size in each region according to the criteria of OBEC. These school sizes are classified as follows: 1) small size of 0-499 students, 2) medium size of 500-999 students, 3) large size of 1,000-1,499 students, and 4) extra-large size of 1,500 or more students.

Samples of this study used quota sampling methods from the four regions of Thailand totaling twenty-four teachers, which includes three teachers from the North, eleven teachers from the Northeast, seven teachers from the Central region, and three teachers from the South.

As shown in Table 1, teachers were chosen based on the following criteria:

1. Experience in teaching technology for at least five years and
2. Having experience in learning instruction that promotes computational thinking abilities.

Table 1 Frequency and percentages of samples that were classified by experience in teaching technology

Background		School Sizes				Total
		Small	Medium	Large	Extra-large	
Experience in teaching technology	5 -10 years	3 (30.00%)	1 (100.00%)	0 (0.00%)	3 (33.33%)	7 (29.17%)
	11 – 15 years	2 (20.00%)	0 (0.00%)	1 (25.00%)	3 (33.33%)	6 (25.00%)
	16 - 20 years	3 (30.00%)	0 (0.00%)	2 (50.00%)	1 (11.11%)	6 (25.00%)
	21 - 25 years	0 (0.00%)	0 (0.00%)	0 (0.00%)	1 (11.11%)	1 (4.17%)
	More than 26 years	2 (20.00%)	0 0.00%	1 25.00%	1 11.11%	4 16.67%
Total		**10 (41.67%)**	**1 (4.17%)**	**4 (16.67%)**	**9 (37.50%)**	**24 100.00%**

Research Instruments

Research instruments in this study were surveys and structured interviews that the researcher created. These were qualified by five experts. (Item Objective Congruence: IOC = 0.98)

The experts were chosen based on the following criteria:

1. Two lecturers from the Faculty of Education with experience in computer teaching or having experience in supervising students and at least ten years of professional experience training in computer teaching majors.

2. Two lecturers from the Faculty of Education in the field of Education Technology and Communication who have ten years of expertise and experience of game-based learning, Inquiry-based learning, scaffolding and cognitive tools.

3. One lecturer from the Faculty of Education in the field of Educational Measurement and Evaluation who has ten years of expertise and work experience in measurement and assessment.

Data Collection

Data was collected by surveys and interviews during the months of April 2018 to June 2018.

Data Analysis

We analyzed the data by finding frequency, percentage, mean ($\bar{X}$ and standard deviation (S.D.) in each question. We also included inductive analysis by interpretation and drew conclusions from the information obtained from the interviews.

Results and Discussion

1. The conditions and problems of computational thinking ability in lower secondary students will be discussed in detail via the following points:

1.1 The problems of computational thinking ability.

In this study, we will present the data analysis of problems of computational thinking ability in lower secondary school students according to computational thinking components

Considering the elements of computational thinking, Figure 2 shows that it was found that the top three components with the lowest average score were: problem decomposition and modularity, data practices, and simulation.

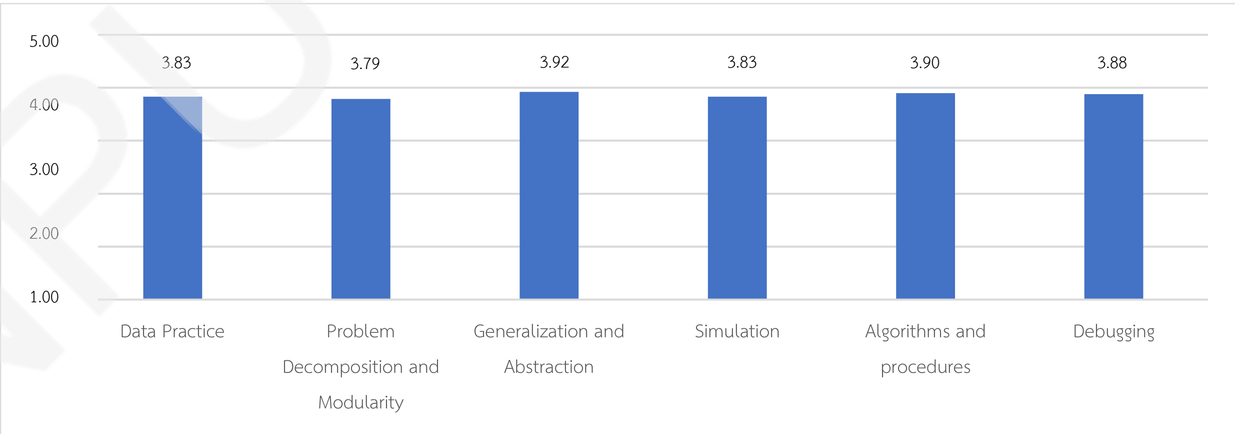

Figure 2 The mean of opinions on the problem of computational thinking in lower secondary students

Considering each minor component of a computational thinking concept the details are as follows:

1.1.1 The three lowest ranked average scores in data practices were: 1) students have critical code reading ability 2) students can identify a concept for the project and 3) students can use research to their advantage while maintaining authenticity.

1.1.2 The three lowest ranked average scores in problem decomposition and modularity were: 1) students can break down the problem into smaller, more manageable parts 2) students can piece together parts of a solution to solve a problem and 3) students can put together smaller parts to make something larger.

1.1.3 The lowest ranked average scores in generalization and abstraction were: 1) students can identify patterns, similarities and connections between prior and current problems and 2) students can make inferences.

1.1.4 The three lowest ranked average scores in simulation were: 1) students can develop a design plan 2) students are comfortable adapting the plan in response to new or different information and 3) students can implement the design plan.

1.1.5 The lowest ranked average scores in algorithms and procedures were: 1) students can create a set of steps to solve a problem and 2) students can solve similar problems with the same set of steps or principles.

1.1.6 The three lowest ranked average scores in debugging were: 1) students can anticipate and plan for a problem 2) students can develop strategies for dealing with problems and 3) students can evaluate a solution to debugging.

1.2 Conditions of teaching and learning that promote computational thinking.

It was found that, overall, teachers had opinions on conditions of computational thinking instruction in the present at the 61-80% operational level. Most teachers organized activities using the inquiry learning process at the 61-80% operational level ($\bar{X}$=4.38), followed by scaffolding at the 61-80% operational level ($\bar{X}$=4.29), and computational thinking at the 61-80% operational level ($\bar{X}$=4.02). On the other hand, it was found that the least used were visual programming tools which were at the 41-60% operational level ($\bar{X}$=3.25) followed by game-based learning at the 41-60% operational level ($\bar{X}$=3.44) as shown in Figure 3.

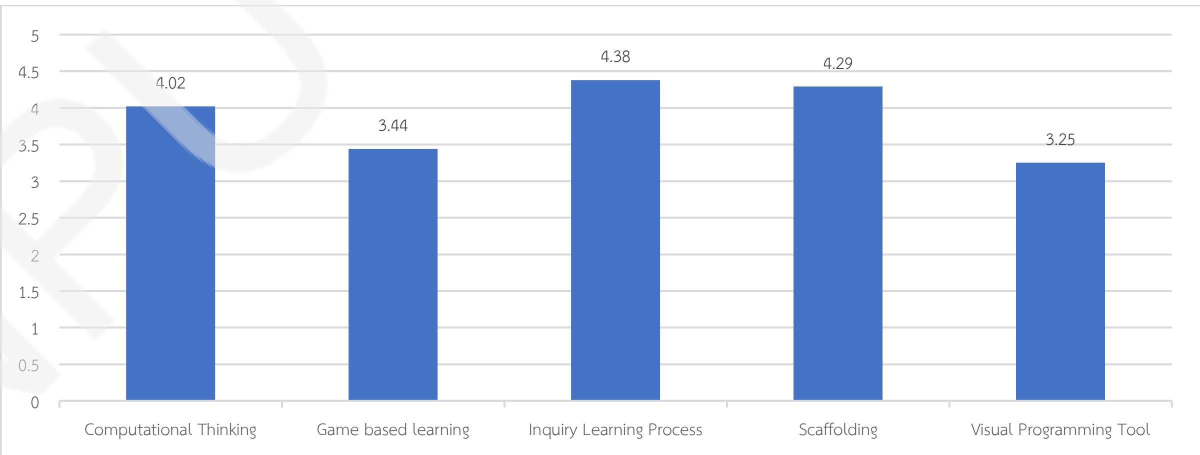

Figure 3. The mean of opinions on conditions of computational thinking instruction

1.3 Factors affecting students' comprehension of computational thinking.

As shown in Table 2, according to the teachers' responses, it was found that there were 45 data that reflect factors affecting students' comprehension of computational thinking. The data was organized into four groups of factors which were:

1. Students; such as their mental readiness for class, their interests, their curiosities about the school subjects, their basic knowledge of mathematics, science, and English, as well as their problem-solving experiences.

2. Teachers; such as their professional knowledge, ability to pass on knowledge, preparation of teaching materials and learning activities, their use of questions, their personality, etc.

3. Learning activities that enable students' understanding and motivation; such as competitions and group activities.

4. Other factors; such as availability of equipment, school administration, budget, instructional media, etc.

The analysis results presented by frequency and percentage found that the factors most affecting students' comprehension of computational thinking were students (48.89%), followed by teachers (28.89%), then by other factors (17.78%), and finally by learning activities (4.44%).

Table 2 Frequency and percentage of factors affecting students' comprehension of computational thinking

Factors affecting students' comprehension of computational thinking	Frequency	%
Students	22	48.89
Teachers	13	28.89
Learning Activities	2	4.44
Other	8	17.78
Total	**45**	**100.00**

1.4 Methods to prepare students

As shown in Table 3, according to the teachers' answers, it was found that there were 52 data that reflect methods to prepare students. The data was organized into three groups of methods which were:

1. Analyzing the content, students, and purposes; such as content analysis: breakdown of the problems and contents, sequencing the difficulty of content, problem solving, practice, objective analysis, as well as analysis of students' abilities in science and mathematics.

2. Preparation of media and equipment; such as software, equipment, media, worksheets, practice exercises, simulations, problems, etc.

3. The design of learning activities; such as inspiration, showing the value of studying, stimulating interest in the subject matter, reviewing, asking questions, teaching introductory courses, teaching about thinking skills, how to think step by step, virtual programming tools, game-based learning, how to teach a friend, informing the students about what content will be studied at a later time so students can prepare, teaching CT in the context of daily life, etc.

The analysis results presented by frequency and percentage found that the most frequent methods to prepare students were: the design of learning activities (63.46%), followed by preparation of media and equipment (21.15%), and analyzing the content, students, and purposes (15.38%).

Table 3 Frequency and percentage of methods to prepare students

Methods to prepare students	Frequency	%
Analyzing the content, students, and purposes	8	15.38
Preparation of media and equipment	11	21.15
The design of learning activities	33	63.46
Total	**52**	**100.00**

1.5 Teaching methods, techniques, activities, or strategies that promote computational thinking ability.

As shown in Table 4, according to the teachers' responses, it was found that there were 32 data that reflect teaching methods, techniques, activities, or strategies that promote computational thinking ability. The data was organized into nine groups which were:

1. Inquiry-based learning
2. Problem-based learning
3. Practice or frequent action
4. Game-based learning
5. Group process-based learning
6. Research-based learning
7. Teaching methods by using demonstration
8. Inductive learning
9. STEM

The analysis results presented by frequency and percentage found that teaching methods, techniques, activities, or strategies that promote computational thinking ability the most were: inquiry-based learning (25.00%), followed by practice or frequent action (21.88%), and then problem-based learning (21.88%).

Table 4 Frequency and percentage of teaching methods, techniques, activities, or strategies that promote computational thinking ability

Teaching methods, techniques, activities, or strategies that promote computational thinking ability	Frequency	%
Inquiry-based learning	8	25
Problem-based learning	7	21.88
Practice or frequent action	7	21.88
Game-based learning	4	12.5
Group process-based learning	2	6.25
Research-based learning	1	3.13
Teaching method using demonstration	1	3.13
Inductive learning	1	3.13
STEM	1	3.13
Total	**32**	**100.00**

1.6 Teaching methods, techniques, or strategies used that were the most successful for instruction.

As shown in Table 5, according to the teachers' answers, it was found that there were 32 data that reflect teaching methods, techniques, or strategies that were used. The data showed that the most successful teaching methods, techniques, or strategies for instruction were:

1. Inquiry-based learning
2. Problem-based learning
3. Game-based learning
4. Practice or frequent action
5. Uncategorized (there is no one method that is the most suitable because it depends on the context of the learner.)

The analysis results presented by frequency and percentage found that the teaching methods, techniques, or strategies used that were most successful for instruction were as follows: inquiry-based learning (33.33%), followed by problem-based learning (20.83%), game-based learning (20.83%), practice or action often (12.50%), and finally, uncategorized (there is no one method that is the most suitable because it depends on the context of the learner) (12.50%).

Table 5 Frequency and percentage of teaching methods, techniques, or strategies that were used were most successful for instruction

Teaching methods, techniques, or strategies that were used were most successful in instruction.	**Frequency**	**%**
Inquiry-based learning	8	33.33
Problem-based learning	5	20.83
Game-based learning	5	20.83
Practice or frequent action	3	12.50
Uncategorized (there is no one method that is the most suitable because it depends on the context of the learner.)	3	12.50
Total	**24**	**100.00**

1.7 Characteristics of the teaching process that emphasize computational thinking.

As shown in Table 6, according to the teachers' responses, it was found that there were 57 data that reflect the characteristics of the teaching process that emphasizes computational thinking. The data was organized into five groups which were:

1. Teach essential basic knowledge such as the fundamentals of equipment and programs, science, and mathematics etc.
2. Watch a sample video clip such as about automated robots to encourage motivation.
3. Learn from diverse problem situations, analyze problems, break down problems into smaller problems, or solve problems by trial-and-error by working together as a group to form conclusions.
4. Use inquiry-based learning by asking motivating questions to find answers.
5. Use of technological processes to solve problems (including frequent practice sessions) whereby students plan, design, and implement

The analysis results presented by frequency and percentage found that the characteristics of the teaching process which emphasized computational thinking most which were as follows: learn from diverse problem situations, analyze problems, break down problems into smaller problems, solve problems by trial-and-error by working together as a group to form conclusions (54.39%), followed by the use of technological processes to solve the problems (24.56%), then utilize inquiry-based learning (10.53%), followed by teach essential basic knowledge (7.02%), and finally, watch a sample video clip to encourage motivation (3.51%).

Table 6 Frequency and percentage of the teaching process that emphasizes computational thinking

The teaching process that emphasizes computational thinking.	Frequency	%
Learning from diverse problem situations	31	54.39
Use of technological processes to solve the problems	14	24.56
Inquiry-based learning	6	10.53
Teaching essential basic knowledge	4	7.02
Watching a sample video clip to encourage motivation	2	3.51
Total	57	**100.00**

1.8 Methods of sharing knowledge.

As shown in Table 7, according to the teachers' answers, it was found that there were 33 data that reflect methods of sharing knowledge. The data was organized into three groups which were:

1. Sharing knowledge between students within the classroom / school, such as presenting in the classroom, sharing knowledge in groups, organizing an exhibition, etc.

2. Sharing knowledge between students through social networks including discussion groups through websites, applications, Facebook Messenger, Facebook, LINE, YouTube, etc.

3. Sharing knowledge of students through learning management systems (LMS) such as Google Classroom.

The analysis results presented by frequency and percentage found that the most common method of sharing knowledge were as follows: sharing knowledge of students through social networks (57.58%), followed by sharing knowledge of students within the classroom / at school (39.39%), and finally, sharing knowledge of students through LMS (3.03%).

Table 7 Frequency and percentage of methods of sharing knowledge

Methods of sharing knowledge.	Frequency	%
Sharing knowledge of students within the classroom / school	13	39.39
Sharing knowledge of students through social networks	19	57.58
Sharing knowledge of students through LMS	1	3.03
Total	**33**	**100.00**

From the above findings, it can be concluded that there were problems about computational thinking instruction with 1) problem decomposition and modularity, 2) data practices, and 3) simulation. Furthermore, there were four factors that enabled students' comprehension in computational thinking as follows: students, teachers, learning activities and other findings. The four factors revealed that teachers should analyze contents, students, and objectives, prepare instructional media and equipment, and design learning activities before teaching to prepare students to be ready to study. These results underline the importance of the computational thinking method, especially in inquiry-based learning, project-based learning, game-based learning, and group process-based learning. This is consistent with a finding by Tikva and Tambouris (2021) that divided learning strategies into six groups as follows: game-based related strategies, modeling and simulations-based related strategies, problem solving related strategies, project-based related strategies, scaffolding related strategies and collaborative related strategies. It is also consistent with a finding by Lyon and Magana (2021) that suggested teaching and learning principles includes the following: 1) creating a model which helps students design and simulate knowledge with the use of programming knowledge and to apply this knowledge to real-world experiences; 2) providing tools like diagrams, equations, flow charts, algorithms and computational models as scaffolding to help students who cannot create and connect multiple forms by themselves; 3) suggesting different perspectives for students to have a chance to explore various solutions; 4) creating opportunities for students to share their knowledge and to compare their models with other models; and 5) challenging students to defend their thinking, assumptions, limitations, and reasoning processes from beginning to end and to document their findings.

In addition, the findings found that most teachers use social networks to share their students' knowledge. This is consistent with the finding by Robles Moral, Fernández Díaz and Aguaded (2021) that claimed social networks affected students' knowledge sharing intentions within and outside the teams. Students can share knowledge within their team rather than outside the team by quickly interacting with many other members. As a member in the network, students can actively share knowledge with each other and receive the benefits that they have by being in a network.

2. Directions for developing a learning model should consist of these six main steps:

2.1 Analyze contents, students, purpose, preparation of instructional media and design of learning activities; understand the need to study to stimulate interest in the subject matter and the use of questions; ensure that incoming students have the required knowledge to prepare them before teaching the new material.

2.2 Choose the teaching method that is most suitable for the students to promote effective computational thinking.

2.2.1 Choose teaching methods, techniques, or strategies that focus on child-centered learning.

2.2.2 Choose computational thinking practice to suit the students' abilities by organizing problems in daily life ranging from easy to difficult.

2.3 Create computational thinking experiences for students both in the classroom and outside the classroom.

2.3.1 Encourage students to have continuous and consistent experiences in computational thinking by practicing or doing frequent actions of various problems, as well as practicing thinking, analyzing, and solving problems, and drawing conclusions in addition to presenting learning outcomes through lectures, compare-and-contrast analysis of advantages and limitations, and helping students gain hands-on experience.

2.3.2 Schools should promote and support projects on computational thinking or projects that can be integrated into the computational thinking course to provide a platform for students to present their work effectively.

2.4 Create a community between teachers and students in their respective school communities as well as communities outside their own school community such as private schools or schools in other school districts. For example, by organizing academic seminars together or organizing a joint competition once a year to share and learn experiences about computational thinking.

2.5 Teachers should improve their computational thinking continually. They should study teaching methods regarding computational thinking from books or online media that is readily available.

2.6 Promote research in computational thinking that encourages teachers to invent new teaching methods to develop current teaching and learning practices that emphasize computational thinking abilities.

From the above findings, it can be concluded that there are six steps for developing a learning model. The first step encompasses analyzing contents, students, purpose, preparation of instructional media and design of learning activities. The second step is choosing the teaching method that is suitable for the students. The third step is creating computational thinking experiences for students both in the classroom and outside the classroom. The fourth step is creating a community between teachers and students in their respective school communities as well as communities outside their school community such as private schools or schools in another school districts. The fifth step is teachers developing their own computational thinking continually. The final step is teachers promoting computational thinking research. This is consistent with Yadav et al. (2017) who stated that there was a need to prepare teachers to understand the importance of computational thinking within their own teaching. They went on to say that there was a need for professional development workshops to inspire teachers to include computational thinking into their specific subjects in lieu of generic workshops that were not specific but only general. Finally, they stated that implementing computational thinking should be included with research that examines whether CT activities had the desired include on student outcomes. It is also consistent with the findings by Lyon and Magana (2021) which suggested that CT in our design-based research intervention can be learned via specific principles which could be beneficial as students generate artifacts in association with building a computational model. The three principles were suggested are the following: 1. Create modeling and simulation experiences that combine engineering disciplinary knowledge with programming knowledge in the context of real-world experiences. 2. Provide scaffolding to help students create and connect multiple forms of representations such as diagrams, equations, flow charts, algorithms, and computational models. 3. Provide opportunities to explore multiple approaches and solutions and provide guidance to consider other perspectives. 4. Orchestrate participant structures by allowing students to share knowledge and to compare and contrast their models. 5. Prompt students to explain their thinking, assumptions, limitations, and reasoning processes throughout the model building process and documentation.

Conclusion

The finding showed that teachers agreed that there are problems in computational thinking in problem decomposition and modularity. There are four factors affecting students' comprehension of computational thinking: 1) students; 2) teachers; 3) learning activities; and 4) other factors. Methods to prepare students are firstly, analyzing the content, students, purposes; secondly, preparation of media and equipment and thirdly, the design of learning activities. Methods of sharing knowledge are firstly, in the classroom or at school; secondly, through social networks; and thirdly, through Learning Management Systems (LMS). From these findings, it is suggested that directions for developing a learning model consist of analyzing contents, students, purpose, preparation of instructional media and design of learning activities. Choose a teaching method that is suitable for the student. Create computational thinking experience for students both in the classroom and outside the classroom. Create a community about computational thinking. The teachers should develop their computational thinking continually and should also promote computational thinking research.

Contribution

Overall, it should be noted that this paper contributes to the provision of information for teachers to develop and improve teaching methods and models of computational thinking.

Suggestion

In this paper, we suggested that the directions to develop a learning model that promotes computational thinking ability were for lower secondary students. If you want to use these directions for another population such as kindergarten students, middle schoolers, senior high school students or university students be sure to apply the teaching methods that suit the age group. Further work might expand by using different student populations such as kindergarten students, middle schoolers, senior high school students or university students and applying a procedure in a model suitable for that age group. In addition, there should be continuing education about students' learning achievements to monitor and assess students' progress and to improve the teaching and learning management for maximum efficiency.

Limitations

The results of this study are limited to the sampling and setting where the study took place and should not be generalized to students in other contexts. We collected data about conditions and problems of computational thinking instruction of Thai lower secondary students with Thai teachers only. If you want to use the results of this study, you must consider this restriction.

References

Amnuayporn, P., Yampinij, S., Meejaleurn, S., and Supaluk, S. (2019). Developing the Media of Game Based Learning Using Gamification Strategy to Enhance Computational Thinking on Computing Science Subject for Students in Primary Year 2. *Journal of Education, Faculty of Education, Srinakharinwirot University.* 20(2),68-78.

Asbell-Clarke, J., Rowe, E., Almeda, V., Edwards, T., Bardar, E., Gasca, S., Baker, R. S., Scruggs, R. (2021). The development of students' computational thinking practices in elementary- and middle-school classes using the learning game, Zoombinis. *Computers in Human Behavior. 115,*106587.

Basu, S., Biswas, G., and Kinnebrew, J. (2017). Learner modeling for adaptive scaffolding in a Computational Thinking-based science learning environment. *User Modeling and User-Adapted Interaction.* 27(1),5-53.

Brasiel, S., Close, K., Jeong, S., Lawanto, K., Janisiewicz, P., and Martin, T. (2017). Measuring Computational Thinking Development with the FUN! Tool. In P. J. Rich and C. B. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 327-347). Cham : Springer International Publishing.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., and Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. *Review of Educational Research.* 87(4),834-860.

Caeli, E. N., and Yadav, A. (2020). Unplugged Approaches to Computational Thinking: a Historical Perspective. *TechTrends.* 64(1),29-36.

Daungjun, S. (2018). Effects of Using Stem Education in Physics On Computational Thinking Ability of Upper Secondary School Students. Master's thesis. Bangkok : Chulalongkorn University.

Eggen, P. D., and Kauchak, D. P. (2011). *Strategies and models for teachers : teaching content and thinking skills*. 6th ed. Boston : Pearson/Allyn and Bacon.

Eggen, P. D., and Kauchak, D. P. (2016). *Educational Psychology: Windows on Classrooms.* 10th ed. New Jersy : Pearson.

Flórez, F. B., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., and Danies, G. (2017). Changing a Generation's Way of Thinking: Teaching Computational Thinking Through Programming. Review of Educational Research. 87(4),834–860.

Gleasman, C., and Kim, C. (2020). Pre-Service Teacher's Use of Block-Based Programming and Computational Thinking to Teach Elementary Mathematics. *Digital Experiences in Mathematics Education, 6* (April 2020),52–90.

Hannafin, M. J., Land, S., and Oliver, K. (1999). Open learning environments: Foundations, Methods, and Models. In C. M. Reigeluth (Ed.), *Instructional-design theories and models Volume II: A new paradigm of Instructional theory*. London : Lawrence erlbaum associates.

Haruehansawasin, S., & Kiattikomol, P. (2018). Scaffolding in problem-based learning for low-achieving learners. *Journal of Educational Research*. 111(3),363–370.

Hickmott, D., Prieto-Rodriguez, E., and Holmes, K. (2018). A Scoping Review of Studies on Computational Thinking in K–12 Mathematics Classrooms. *Digital Experiences in Mathematics Education. 4*(1),48-69.

Hsu, T.-C., Chang, S.-C., and Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education.* 126,296-310.

International Society for Technology in Education. (2017). ISTE Standards for Students. Retrieved Febuary 2021, from https://www.iste.org/standards/for-students.

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., and Grise, K. (2018). Computational What? Relating Computational Thinking to Teaching. *TechTrends.* 62(6),574-584.

Karl, B. (2017). *Computational Thinking : A Beginner's Guide to Problem-solving and Programming*. Swindon, UK : BCS, The Chartered Institute for IT.

Khamnaen, N. (2019). Development of A Flipped Learning Model with Problem-Based Learning and Gamifications to Enhance Computational Thinking and Achievement Motivation of Upper Secondary School Students. Master's thesis. Bangkok : Chulalongkorn University.

Korkmaz, Ö., Çakir, R., and Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior.* 72(2017),558-569.

Lawanto, K., Close, K., Ames, C., and Brasiel, S. (2017). Exploring Strengths and Weaknesses in Middle School Students' Computational Thinking in Scratch. In P. J. Rich and C. B. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 307-326). Cham : Springer International Publishing.

Lee, I., Grover, S., Martin, F., Pillai, S., and Malyn-Smith, J. (2020). Computational Thinking from a Disciplinary Perspective: Integrating Computational Thinking in K-12 Science, Technology, Engineering, and Mathematics Education. *Journal of Science Education and Technology.* 29(1),1-8.

Lyon, J. A., and Magana, A. J. (2021). The use of engineering model-building activities to elicit computational thinking: A design-based research study. *Journal of Engineering Education*. 110(1),184-206.

Malyn-Smith, J., and Angeli, C. (2020). Computational Thinking. In A. Tatnall (Ed.), *Encyclopedia of Education and Information Technologies* (pp. 333-340). Cham : Springer International Publishing.

Office of the Basic Education Commission. (2017). Learning Standards and Indicators of the Basic Education Core Curriculum in Science B.E. 2551 (Revised 2017). Bangkok : Agricultural Co-operative Federation of Thailand, Ltd.

Office of the Education Council. (2017). The National Education Plan (2017-2036). Bangkok : Prikwan Graphic.

Repenning, A., Basawapatna, A. R., and Escherle, N. A. (2017). Principles of Computational Thinking Tools. In P. J. Rich and C. B. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 291-305). Cham : Springer International Publishing.

Robles Moral, F. J., Fernández Díaz, M., and Aguaded, I. (2021). Future Primary School Teachers' Digital Competence in Teaching Science through the Use of Social Media. *Sustainability (2071-1050).* 13(5),2816.

Rose, S. P., Habgood, M. P. J., and Jay, T. (2017). An Exploration of the Role of Visual Programming Tools in the Development of Young Children's Computational Thinking. *Electronic Journal of e-Learning,* 15(4),297-309.

Saputri, A. A., and Wilujeng, I. (2017). Developing Physics E-Scaffolding Teaching Media to Increase the Eleventh-Grade Students' Problem Solving Ability and Scientific Attitude. *International Journal of Environmental and Science Education.* 12(4),729-745.

Song, D., Hong, H., and Oh, E. Y. (2021). Applying computational analysis of novice learners' computer programming patterns to reveal self-regulated learning, computational thinking, and learning performance. *Computers in Human Behavior.* 120(2021),106746.

Supaluk, S. (2018). A Cloud Based Learning System Using Reverse Engineering Approach and Peer to Peer Technique to Enhance Computational Thinking. Doctoral dissertation. Bangkok : Chulalongkorn University.

Teemueangsa, S., and Jedaman, P. (2021). Developing science learning activities using online gamification. Paper presented at the Journal of Physics : Conference Series.

Threekunprapa, A., and Yasri, P. (2020). Patterns of computational thinking development while solving unplugged coding activities coupled with the 3s approach for self-directed learning. *European Journal of Educational Research.* 9(3),1025-1045.

Tikva, C., and Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers and Education. 162*(March 2021),104083.

Vygotsky, L. S. (1979). The Development of Higher Forms of Attention in Childhood. *Soviet Psychology.* 18(1),67-115.

Yadav, A., Good, J., Voogt, J., and Fisser, P. (2017). Computational Thinking as an Emerging Competence Domain. In M. Mulder (Ed.), *Competence-based Vocational and Professional Education: Bridging the Worlds of Work and Education* (pp. 1051-1067). Cham : Springer International Publishing.

Yildiz Durak, H. (2020). The Effects of Using Different Tools in Programming Teaching of Secondary School Students on Engagement, Computational Thinking and Reflective Thinking Skills for Problem Solving. *Technology, Knowledge and Learning.* 25(1),179-195.