A Conceptual Review of Assessing Problem-Solving Skills through Computational Thinking

Thipaphan Boonmee[1] and Krittayakarn Topithak[2]
Naresuan University, Thailand
Corresponding Author, Email: [1]thipaphanb64@nu.ac.th

Retrieved: September 14, 2024; **Revised:** July 3, 2025; **Accepted:** August 5, 2025

Abstract

This article presents concepts, methods, and future directions for assessing problem-solving skills through computational thinking, a crucial 2st century skill. The content covers the theoretical foundations of computational thinking, assessment frameworks, evaluation methods and tools, data analysis and interpretation, challenges and limitations, practical applications of assessment results, and future trends and innovations. The article emphasizes the importance of developing effective assessment tools to support learners in the digital age and provides recommendations for educational institutions to integrate computational thinking into curricula. Furthermore, it analyzes current practices and emerging innovations, making a significant contribution to advancing assessment approaches for problem-solving skills through computational thinking, with particular emphasis on the Thai educational context.

Keywords: Computational Thinking; Skill Assessment; Problem-Solving; Digital Age Education

Introduction

In an era characterized by rapid technological change and innovation, developing learners' ability to adapt to these changes is a priority across all sectors. This is particularly true for the development of problem-solving skills, as they have become one of the most critical skills for living and working dynamic and unpredictable environment of the 21st century. The ability to identify problems, analyze their root causes, and systematically devise solutions is viral for promoting lifelong learning and adaptation in a complex society (Panich, 2012).

The importance of problem-solving skills in education and daily life is widely acknowledged. The World Economic Forum (2020) has identified complex problem-solving as one of the most crucial skills for future workforce, as modern labor market demands personnel who can handle unexpected challenges and problems without ready-made solutions. Furthermore, OECD studies (2018) indicate that problem-solving skills are a significant factor and a key determinant of success in learning and work.

In the Thai context, the Office of the Basic Education Commission (OBEC) has designated problem-solving skills as one of the key competencies for learners in the Basic Education Core Curriculum B.E. 2551 (2008) (revised edition B.E. 2560 (2017)). It emphasizes enabling learners to solve problems and overcome various obstacles appropriately, based on reason, ethics, and information (Ministry of Education, 2017). The Institute for the Promotion of Teaching Science and Technology (IPST) initiated the development of a Computing Science curriculum and learning management program in 2017, focusing on developing computational thinking skills alongside computer science knowledge (Institute for the Promotion of Teaching Science and Technology, 2018).

This article aims to present a comprehensive overview of concepts, methods, and future directions for assessing problem-solving skills through computational thinking. It analyzes current practices, challenges, and emerging trends, with the goal of informing the development of computational thinking integration into educational curricula and assessment frameworks.

Theoretical Foundations of Computational Thinking

Computational thinking is a concept widely disseminated by Jeannette Wing (2006), who described it as a thought process involving problem formulation and solution representation in a form that can be effectively carried out by an information-processing agent. This concept is not limited to computer programming but extends to systematic thinking and problem-solving applicable across various disciplines.

Components of Computational Thinking

Grover and Pea (2013) identified four key components of computational thinking:

1. Decomposition: This involves breaking down complex problems into smaller, more manageable parts. This method helps us understand and solve seemingly difficult problems more easily.

Educational Setting Example: In a 5th-grade mathematics classroom, when students encounter a word problem about planning a school fair, teachers can guide them through decomposition by asking: "What are all the different things we need to figure out?" Students might identify: calculating booth space, determining ticket prices, estimating attendance, and managing volunteer schedules. Each component can then be tackled separately, making the overall problem less overwhelming.

Additional Examples:

- In application development, we might divide the work into sections such as user interface design, database development, and coding for various functions.
- In planning large projects, we might break down the work into smaller activities and assign responsibilities for each part.

Benefits:

- Makes complex problems appear simpler and more manageable
- Aids in planning and prioritizing tasks
- Facilitates teamwork through clear task division

2. Pattern Recognition: This is the ability to identify similarities or patterns in data or problems. This skill helps us use existing experience and knowledge to solve new problems efficiently.

Educational Setting Example: In a science classroom studying weather patterns, students can analyze daily temperature and precipitation data over several months. They learn to identify seasonal patterns, recognize that certain cloud formations often precede rain, and notice that temperature changes follow predictable daily cycles. This pattern recognition then helps them make weather predictions for upcoming weeks.

Additional Examples:

- In programming, we might notice similar work patterns and create reusable functions.
- In financial data analysis, we might find recurring patterns in stock price changes.

Benefits:

- Assists in predicting and forecasting future events
- Increases problem-solving efficiency by using previously successful methods
- Promotes model creation and data analysis

3. Abstraction: This is the process of identifying and isolating the essential aspects of a problem or system, discarding unnecessary details. This skill helps us create models or representations that are easy to understand and manage.

Educational Setting Example: When teaching students about ecosystems, teachers can use abstraction by creating simplified food web diagrams that show only the most important feeding relationships, excluding minor species and complex interactions. Students learn to focus on key concepts like producers, primary consumers, and apex predators, making the ecosystem easier to understand before adding complexity.

Additional Examples:

- A subway map that shows only stations and routes without considering actual distances or topography.
- Creating classes in object-oriented programming, which capture only the essential properties and behaviors of objects.

Benefits:

- Reduces problem complexity and makes it easier to understand
- Helps communicate complex ideas more simply
- Facilitates the creation of models and systems applicable across various disciplines

4. Algorithmic Thinking: This is the process of developing a clear set of instructions or rules to solve a problem or achieve a goal. This skill involves thinking in a step-by-step and systematic manner.

Educational Setting Example: In an elementary classroom, students learning long division can practice algorithmic thinking by creating step-by-step instructions for solving division problems. They might write: "1) Estimate how many times the divisor goes into the first digits,

2) Multiply and subtract, 3) Bring down the next digit, 4) Repeat until complete." This algorithm can then be applied to any long division problem.

Additional Examples:

- Writing a detailed recipe that specifies cooking steps.
- Designing the workflow of an automated system in a factory.

Benefits:

- Enables systematic and efficient problem-solving
- Facilitates the creation of computer programs and automated systems
- Promotes logical and sequential thinking

Alignment with Learning Theories

Computational thinking aligns with several established learning theories, providing a strong theoretical foundation for its incorporation into teaching and learning. This alignment demonstrates how computational thinking naturally supports effective pedagogical practices:

1. Constructivism Theory by Piaget and Vygotsky

Constructivism emphasizes that learners actively construct knowledge through experience and interaction with their environment, rather than passively receiving information. Computational thinking strongly supports constructivist learning principles in several ways:

- Active Knowledge Construction: When students engage in computational thinking activities, they build mental models through hands-on problem-solving experiences. For example, when decomposing a complex problem, students construct their understanding of how large problems relate to smaller components.

- Social Construction (Vygotsky's Zone of Proximal Development): Computational thinking activities often involve collaborative problem-solving, where students work together to break down problems, identify patterns, and develop algorithms. More experienced peers or teachers can scaffold learning by guiding students through computational thinking processes.

- Experiential Learning: Students learn computational thinking concepts not through memorization but through practical application. When they debug a program or refine an algorithm, they construct new understanding based on trial, error, and reflection.

- *Practical Application in Education:* Teachers can design computational thinking activities where students build robots or create simple games, allowing them to construct knowledge about programming concepts, logical thinking, and problem-solving strategies through direct experience (Piaget, 1952; Vygotsky, 1978).

2. Multiple Intelligences Theory by Howard Gardner

Gardner's theory proposes that humans possess multiple types of intelligence, and computational thinking engages several of these intelligences simultaneously:

- Logical-Mathematical Intelligence: This is the most obvious connection, as computational thinking heavily involves logical reasoning, pattern recognition, and systematic problem-solving approaches.

- Spatial Intelligence: When students create visual representations of algorithms (like flowcharts) or design user interfaces, they engage spatial reasoning skills.

- Linguistic Intelligence: Writing clear algorithms and communicating computational solutions requires precise language use and clear expression of logical sequences.

- Interpersonal Intelligence: Many computational thinking projects involve teamwork, requiring students to collaborate, share ideas, and build on others' contributions.

- Educational Implications: This alignment suggests that computational thinking can be an inclusive pedagogical approach that engages learners with different intellectual strengths, not just those who excel in traditional mathematical or scientific domains (Gardner, 1983).

3. Meaningful Learning Theory by Ausubel

Ausubel's theory emphasizes the importance of connecting new knowledge with learners' existing knowledge structures. Computational thinking facilitates meaningful learning by:

- Building on Prior Knowledge: Students can apply computational thinking to problems in domains they already understand (like organizing their daily schedule or planning a party), making new computational concepts more accessible.

- Creating Cognitive Bridges: Computational thinking concepts like algorithms can be connected to familiar processes (like following a recipe), helping students build meaningful connections between abstract computational concepts and concrete experiences.

- Promoting Transfer: When students learn to decompose problems in one context (like mathematics), they can transfer this skill to other domains (like science experiments or creative writing projects).

- Classroom Implementation: Teachers can introduce computational thinking by starting with familiar problems and gradually introducing more abstract computational concepts, ensuring students can connect new learning to their existing knowledge base (Ausubel, 1963).

4. Information Processing Theory

This theory explains the brain's processes of receiving, storing, and processing information, which directly relates to computational thinking processes:

- Attention and Working Memory: Decomposition helps manage cognitive load by breaking complex problems into manageable chunks that don't overwhelm working memory capacity.

- Pattern Recognition and Long-term Memory: The pattern recognition component of computational thinking helps students organize information in long-term memory by identifying recurring structures and relationships.

- Executive Processing: Algorithmic thinking develops students' executive function skills, including planning, sequencing, and monitoring problem-solving processes.

- Information Organization: Abstraction skills help students organize and categorize information efficiently, focusing attention on relevant details while filtering out distracting information.

- Cognitive Benefits:Understanding these connections helps educators design computational thinking activities that support students' natural information processing capabilities while gradually developing more sophisticated cognitive skills (Miller, 1956).

5. Integration Benefits

The convergence of these learning theories in computational thinking suggests several pedagogical advantages:

- Holistic Development: Computational thinking addresses multiple aspects of learning simultaneously - cognitive, social, and metacognitive.
- Inclusive Learning: Different theoretical perspectives support different learning styles and intellectual strengths.
- Transferable Skills: The theoretical foundation supports the development of skills that transfer across disciplines and contexts.
- Scaffolded Learning: The theories provide frameworks for progressively building computational thinking capabilities.

Integrating computational thinking into teaching and learning based on these learning theories will enhance the effectiveness of developing learners' problem-solving skills by providing multiple pathways to understanding and multiple ways to demonstrate competence.

The Importance of Applying Computational Thinking to Develop Problem-Solving Skills

In the context of Thailand, applying computational thinking to develop students' problem-solving skills is crucial for several reasons:

1. Enhancing Competitiveness: Students equipped with computational thinking skills will be better prepared for learning and working in the digital age, which will help increase the country's long-term competitiveness (Office of the National Economic and Social Development Council, 2021). In an era where technology plays a crucial role in all sectors, having computational thinking skills will enable Thai workers to adapt and innovate efficiently.

2. Improving Academic Achievement: Thailand's PISA scores in problem-solving remain below the OECD average (OECD, 2019). Incorporating computational thinking into problem solving skill development may help elevate Thai students' academic performance, particularly in mathematics and science, which require analytical and systematic problem-solving skills.

3. Reducing Educational Disparities: The widespread implementation of computational thinking in the Thai education system may help bridge the educational gap between urban and rural schools by providing all students access to essential skills for the digital age (Office of the Education Council, 2020). Developing computational thinking skills doesn't necessarily require expensive technological equipment but can be taught through "unplugged" activities that don't use computers.

4. Preparing for Thailand 4.0: The Thailand 4.0 policy emphasizes economic development through innovation and technology. Developing computational thinking skills will prepare Thai students to compete in the digital era (Office of the National Economic and Social Development

Board, 2017). These skills will enable learners to analyze, design, and develop solutions that address the needs of future society and economy.

5. Fostering Innovation and New Entrepreneurs: Computational thinking skills will promote innovation and entrepreneurship, aligning with the Startup Thailand policy and creative economy development (National Innovation Agency, 2019). Computational thinking helps learners identify business opportunities, analyze problems, and develop solutions that effectively respond to market demands.

Framework for Assessing Problem-Solving Skills through Computational Thinking

An effective assessment of problem-solving skills through computational thinking requires a multidimensional approach that considers various aspects of learner development. A comprehensive assessment framework should include the following dimensions:

1. Knowledge Dimension This dimension relates to understanding the principles and concepts of computational thinking, including:

- Knowledge of the main components of computational thinking (decomposition, pattern recognition, abstraction, and algorithmic design)
- Understanding of basic computer science principles such as Boolean logic, data structures, and basic programming
- Knowledge of applying computational thinking in various fields

Assessment in this dimension can be conducted through knowledge tests, concept presentations, or explanations of various computational thinking principles.

2. Cognitive Process Dimension This dimension focuses on the ability to analyze problems, synthesize solutions, and evaluate outcomes, comprising:

- Ability to analyze and decompose problem elements
- Skills in modeling and abstract thinking
- Ability to design and develop algorithms for problem-solving
- Skills in evaluating and improving problem-solving methods

Assessment in this dimension can be conducted through problem-solving tasks, project design, or case study analysis.

3. Application Dimension This dimension relates to the ability to apply computational thinking skills in real-world situations, including:

- Ability to identify real-life problems that can be solved with computational thinking
- Skills in applying computational thinking tools and techniques in various contexts
- Ability to transfer knowledge and skills to new situations

Assessment in this dimension can be conducted through project work, problem-solving in simulated situations, or internships in real work environments.

4. Social Skills Dimension This dimension emphasizes the ability to work with others and communicate ideas, comprising:

- Teamwork skills in solving complex problems

- Ability to effectively communicate ideas and problem-solving methods
- Leadership and project management skills

Assessment in this dimension can be conducted through observation of group work behavior, project presentations, or peer evaluations.

5. Attitudinal Dimension This dimension relates to learners' attitudes and feelings towards computational thinking and problem-solving, including:

- Confidence in facing complex problems
- Persistence and effort when facing challenges
- Enthusiasm for learning and developing new skills
- Positive attitudes towards using technology in problem-solving

Assessment in this dimension can be conducted through attitude questionnaires, interviews, or long-term behavior observation.

Considering these five dimensions in assessing problem-solving skills through computational thinking will provide a comprehensive overview of learners' abilities. However, these dimensions are not entirely separate but interrelated in various ways:

1. Hierarchical Relationship: Basic skills, such as decomposition, form the foundation for higher- level skills like algorithmic design. Assessment should consider the sequence of skill development.

2. Integrative Relationship: Various skills are often used together in solving complex problems. Assessment should consider the integration of different skills.

3. Complementary Relationship: Some skills enhance the efficiency of others. For example, abstract thinking aids in designing efficient algorithms. Assessment should consider how skills complement each other.

4. Continuous Relationship: Skill development is an ongoing process. Assessment should continuously monitor progress and adapt according to increasing ability levels.

5. Context-Specific Relationship: The importance of different skills may vary depending on the problem context. Assessment should consider the context of real-world application.

Understanding the relationships between these dimensions will help design comprehensive and effective skill assessment tools, considering both the complexity of skills and their application in real situations.

Methods and Tools for Assessing Computational Thinking Skills

Assessing computational thinking skills can be accomplished through various methods, each with its own strengths and limitations. Therefore, choosing appropriate assessment methods is crucial. The following methods and tools can be used to assess computational thinking skills:

1. Performance-based Assessments Performance-based assessments are essential tools for measuring the ability to apply knowledge in real situations. Instructors can design tasks that require learners to perform hands-on activities, such as solving mathematical problems,

conducting scientific experiments, or writing computer programs (Institute for the Promotion of Teaching Science and Technology, 2018).

Advantages:

- Can measure skills in applying knowledge to real situations
- Provides opportunities for learners to demonstrate creativity and diverse problem-solving Methods
- Reflects the ability to integrate various skills

Limitations:

- May be time-consuming, especially with large groups of learners
- Scoring can be complex and requires clear assessment rubrics

Application examples:

- Having learners design and develop simple applications to solve everyday problems
- Organizing mathematics problem-solving competitions using computational thinking

2. Project Evaluations Project evaluations are suitable for measuring teamwork skills, planning, and complex problem-solving. Learners work on long-term projects that require knowledge and skills from various domains.

Advantages:

- Can assess computational thinking skills in complex contexts similar to real situations
- Promotes collaborative learning and team-working skills development
- Provides opportunities for learners to demonstrate creativity and specific expertise

Limitations:

- Time-consuming to implement and evaluate
- May result in unequal participation among team members
- Requires continuous monitoring and guidance from instructors

Application examples:

- Having learners develop projects that use technology to solve local community problems
- Organizing innovation project competitions that use computational thinking to address social or environmental issues

3. Observation and Behavior Recording This method is suitable for assessing social skills, collaboration, and communication skills. Instructors can observe learners' behaviors in various situations and systematically record data.

Advantages:

- Can assess skills that cannot be measured by formal tests, such as collaboration skills
- Provides in-depth information about learners' thinking processes and problem-solving Approaches
- Can continuously assess learners' development

Limitations:

- May be subject to observer bias

- Time-consuming for data collection and analysis
- May make learners uncomfortable if they know they are being observed

Application examples:

- Using behavioral observation forms to assess group work during computational thinking problem-solving tasks
- Recording videos of learners' work to analyze thinking processes and problem-solving approaches

4. Digital Portfolios Digital portfolios are tools that help learners collect and present their work in digital format. They can demonstrate development and learning achievements over a period of time.

Advantages:

- Clearly and continuously shows learners' development
- Promotes learners' reflection and self-assessment
- Can collect various types of work, such as code, videos, or project reports

Limitations:

- Requires technological skills to create and manage digital portfolios
- May be time-consuming to evaluate all work in the portfolio
- May create inequality among learners with different technological skills

Application examples:

- Having learners create digital portfolios that collect projects, code, and reflections on using computational thinking
- Using online platforms for learners to share and exchange computational thinking work

5. Using Technology and Online Platforms for Assessment Modern technology and online platforms offer opportunities for more diverse and efficient assessment (World Economic Forum, 2020).

Advantages:

- Can provide immediate feedback and adjust test difficulty levels according to learners' Abilities
- Collects detailed data and allows for rapid analysis
- Can simulate complex situations to test computational thinking skills

Limitations:

- Requires efficient technological infrastructure
- May face data security and privacy issues
- Instructors and learners need a certain level of technological skills

Application examples:

- Using online learning platforms with automatic assessment systems for computational thinking skills

- Using computer games or simulations to assess problem-solving skills in virtual environments

Choosing diverse and appropriate methods and tools for skill assessment that fit the learning context will help ensure comprehensive evaluation and better reflect learners' true abilities. Instructors should consider selecting methods that align with learning objectives and the nature of the skills being measured to ensure efficient assessment that truly benefits learner development.

Analysis and Interpretation of Data from Assessing Problem-Solving Skills through Computational Thinking

Measuring skills is only the first step in the learner development process. Efficient analysis and interpretation of data obtained from skill assessment will help instructors use the information to improve teaching and learning and support learners appropriately.

1. Quantitative and Qualitative Analysis Techniques

Data analysis from skill measurement can be conducted both quantitatively and qualitatively:

1.1 Quantitative Analysis:

-Use basic statistics such as mean, median, and standard deviation to summarize skill measurement results

-Analyze trends and patterns of scores or assessment results over different time periods

- Use correlation analysis to find relationships between various skills

1.2 Qualitative Analysis:

- Analyze content from qualitative data such as observation notes or learner reflections

- Use inductive analysis to find patterns or themes emerging from the data

- Conduct case study analysis to understand individual learner development

Combining both types of analysis will provide a more complete picture of learners' skills and development.

2. Using Learning Analytics to Track Learner Progress

Learning Analytics is an effective tool for monitoring and analyzing learner development. Approaches to using Learning Analytics include:

1. Systematic data collection: Use Learning Management Systems (LMS) or online platforms to continuously collect data on learners' learning and activity engagement

2. Real-time analysis: Use data analysis tools that can process and display results in real-time, allowing instructors to respond to learners' needs promptly

3. Learning outcome prediction: Use machine learning models to predict learning outcomes and learner risks, enabling proactive support

4. Social network analysis: Study patterns of interaction and collaboration among learners in group activities or collaborative learning

5. Learning personalization: Use data from Learning Analytics to customize content and learning activities to suit individual learners' needs and ability levels

6. Presenting Skill Measurement Results in an Easy-to-Understand Format

3. Presenting skill measurement results in an easy-to-understand format is crucial to help learners, parents, and stakeholders comprehend and utilize the information. Presentation approaches include:

1. Use data visualization:
   - Bar charts or line graphs showing the development of various skills over time
   - Radar charts displaying levels of different skills in comparison
   - Infographics summarizing learners' overall skills and achievements
2. Create user-friendly reports:
   - Use easy-to-understand language, avoiding complex technical terms
   - Present information concisely, using short headings and paragraphs
   - Use colors and icons to highlight important information and make reports more engaging
3. Provide concrete recommendations:
   - Clearly identify strengths and areas for improvement
   - Suggest approaches or activities that will help develop skills in areas needing Improvement
   - Highlight examples of success or noteworthy progress
4. Create interactive dashboards:
   - Design online dashboards that users can interact with
   - Allow users to view data from different perspectives based on their interests
   - Update data in real-time to show the latest progress

Challenges and Precautions in Assessing Problem-Solving Skills through Computational Thinking

Assessing problem-solving skills through computational thinking is an important process in evaluating learners' abilities. However, there are several challenges and precautions to consider:

1. Individual differences among learners: Each learner differs in terms of background knowledge, experience, and learning style. Therefore, assessing problem-solving skills through computational thinking must consider this diversity by:

- Designing flexible and diverse assessments to suit all learner groups
- Avoiding direct comparisons between learners, focusing instead on individual skill Development
- Considering environmental factors that may affect assessment results, such as cultural background or learning environment

Approaches to improve assessment efficiency:

- Use various assessment methods such as testing, project work, and behavior observation to cover different learning styles
- Involve learners in setting their own goals and assessment methods
- Provide additional support for learners with special needs or different backgrounds

2. Validity and reliability of measurement tools: High-quality measurement tools are essential for obtaining reliable results. Consider:

- Checking validity to ensure the tool measures the intended skills
- Testing reliability to ensure consistent measurement results
- Continuously improving measurement tools based on feedback and result analysis

Approaches to improve assessment efficiency:

- Conduct pilot testing with a sample group before implementing the tool - Use statistical methods such as factor analysis to check construct validity
- Perform test-retest to check the reliability of the measurement tool

3. Ethics and privacy in data collection: Assessing problem-solving skills through computational thinking often involves collecting personal data. It is important to:

- Maintain confidentiality and privacy of learner data
- Obtain permission before collecting and using data, especially for young learners
- Use data appropriately and fairly, avoiding any use that may negatively affect learners

Approaches to improve assessment efficiency:

- Develop clear privacy policies and data management practices
- Use data encryption technology to protect learners' personal information
- Provide training for relevant personnel on ethical and legal data management

4. Adapting to rapidly changing technology: Technology plays a crucial role in modern assessment of computational thinking problem-solving skills, but it comes with challenges:

- Selecting appropriate and efficient technology for assessing computational thinking problem-solving skills
- Training assessors to use new technologies effectively
- Ensuring technology does not become a barrier for some learner groups, such as those with limited access to digital devices

Approaches to improve assessment efficiency:

- Establish a dedicated team to monitor and evaluate new technologies related to computational thinking skill assessment
- Develop continuous training and development plans for personnel to keep up with technological changes
- Create assessment alternatives that do not rely heavily on technology to ensure equal access

Application of Problem-Solving Skills Assessment Results through Computational Thinking

Effective application of problem-solving skills assessment results through computational thinking can positively impact the education system in multiple dimensions, including curriculum and teaching method development, individual learner support, and preparation for future careers.

1. Improving Curriculum and Teaching Methods Assessment results of problem-solving skills through computational thinking can be used to enhance curriculum and teaching methods by:

- Analyzing learners' strengths and weaknesses to improve content and learning activities
- Identifying skills that most learners lack to supplement or emphasize in the curriculum
- Evaluating the effectiveness of various teaching methods and adapting them to suit learners
- Developing learning materials that better respond to learners' needs

2. Providing Individualized Guidance and Support Assessment results of problem-solving skills through computational thinking help provide targeted support to learners:

- Creating personalized learning plans tailored to each learner's strengths and areas for Improvement
- Offering specific guidance to develop skills that are still lacking
- Proposing supplementary activities or special projects appropriate to the skill level and interests of learners
- Continuously monitoring learners' progress and adjusting support plans as appropriate

3. Linking Computational Thinking Skills to Future Careers Assessment results of problem-solving skills through computational thinking can be used to prepare learners for future careers:

- Analyzing skills in demand in the labor market and linking them to computational thinking skills
- Providing information about careers related to computational thinking skills, such as software developers, data analysts, or artificial intelligence engineers
- Organizing activities or projects that simulate the use of computational thinking skills in real work situations
- Establishing cooperation with industry to allow learners to see the application of skills in real work environments

This study introduces a comprehensive framework for assessing computational thinking skills in education, bridging theory with practical classroom application.

1. Multidimensional Integration Model

The framework presents the first comprehensive model integrating five interconnected assessment dimensions: knowledge, cognitive processes, application, social skills, and attitudes.

This holistic approach moves beyond traditional technical skill measurement to include collaborative and motivational learning contexts.

2. Culturally-Responsive Methodology

Specifically designed for diverse educational contexts, the methodology incorporates "unplugged" assessment techniques that maintain academic rigor while accommodating schools with limited technological infrastructure, ensuring accessibility across socioeconomic contexts.

3. Learning Analytics Integration

The framework pioneers systematic application of learning analytics for computational thinking assessment, enabling real-time, data-driven educational decision-making and personalized instruction based on dynamic progress tracking.

4. Cross-Disciplinary Transfer Model

Establishes systematic approaches for assessing computational thinking skill transfer across mathematics, science, language arts, and social studies, providing evidence for cross-curricular educational value.

5. Developmental Progression Framework

Creates empirically-based progressions mapping how decomposition, pattern recognition, abstraction, and algorithmic thinking skills develop across age groups and educational levels.

6. Technology-Enhanced Validity Framework

Addresses critical challenges of maintaining assessment validity and reliability in digital environments while ensuring equity and accessibility for all learners.

Research Implications

The framework establishes new research directions including:

- Longitudinal impact studies on computational thinking education effects
- Cross-cultural validation across educational systems
- AI-enhanced assessment applications
- Neurocognitive research on computational thinking mechanisms

Educational Impact

For Educators

- Scientifically-grounded assessment tools
- Structured approaches for comprehensive skill evaluation
- Dynamic progress tracking capabilities

For Policymakers

- Empirical evidence for curriculum integration decisions
- Guidelines for educational technology implementation
- Frameworks for teacher professional development

For Students

- Personalized learning experiences
- Developmentally appropriate assessments
- Enhanced computational thinking skill development

This comprehensive framework represents a significant advancement in educational assessment methodology, transforming how computational thinking is evaluated and taught. By providing both theoretical insights and practical tools, it enables evidence-based instruction, supports policy development, and enhances student learning outcomes in preparation for digital age success.

The framework's emphasis on cultural responsiveness, technological accessibility, and multidimensional assessment makes it particularly valuable for diverse educational contexts, ensuring that computational thinking education can be effectively implemented and evaluated across varied socioeconomic and technological environments.

Conclusion

Assessing problem-solving skills through computational thinking is a crucial process in developing learners to be ready for challenges in the digital age. This article has presented concepts, methods, and future directions for assessing these skills, with key points as follows:

1. The importance of computational thinking in developing problem-solving skills: Computational thinking is not only a necessary skill in technology but also a fundamental basis for developing analytical thinking and problem-solving abilities across all disciplines.

2. Comprehensive assessment framework: Effective assessment must cover dimensions of knowledge, cognitive processes, application, social skills, and attitudes, considering the relationships between these dimensions.

3. Diverse methods and tools: Using a variety of assessment methods, including performance-based assessments, project evaluations, behavior observation, and modern technology, will provide comprehensive and reliable data.

4. Efficient data analysis and interpretation: Using both quantitative and qualitative analysis techniques, including Learning Analytics, will help in understanding learners' development in depth.

5. Addressing challenges and precautions: Awareness of individual differences, maintaining validity and reliability of measurement tools, considering ethics and privacy, and adapting to rapidly changing technology are important in developing an effective assessment system.

6. Effective application of assessment results: Results from the assessment should be used to improve curricula, develop teaching methods, provide individual learner support, and link to future career preparation.

Recommendations for developing the assessment of problem-solving skills through computational thinking in Thailand:

1. Policy and curriculum development: The Ministry of Education should develop policies and curricula that integrate computational thinking across all subjects, not limited to computer or technology classes.

2. Teacher and educational personnel development: Provide training and development for teachers to understand computational thinking concepts and effectively assess these skills.

3. Collaboration with the private sector: Promote cooperation between educational institutions and the private sector in developing modern assessment tools and platforms.

4. Research and development: Support research to develop assessment methods suitable for the Thai context and study the impact of computational thinking skill development on academic achievement and future careers.

5. Raising social awareness: Organize activities and public relations to make society aware of the importance of computational thinking skills in the digital age.

In conclusion, developing an effective system for assessing problem-solving skills through computational thinking will be a significant step in enhancing the quality of Thai education and preparing Thai youth to compete on the global stage. Systematically integrating this concept into the education system will help build a strong foundation for sustainable development of Thailand towards the digital era.

References

Ausubel, D. P. (1963). *The psychology of meaningful verbal learning*. New York: Grune & Stratton.

Gardner, H. (1983). *Frames of mind: The theory of multiple intelligences*. New York: Basic Books.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, *42*(1), 38–43.

Institute for the Promotion of Teaching Science and Technology. (2018). *Computing science learning management manual*. Bangkok, Thailand: Institute for the Promotion of Teaching Science and Technology.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81–97.

Ministry of Education. (2017). *Basic education core curriculum B.E. 2551 (2008) (Revised edition B.E.2560 (2017))*. Bangkok, Thailand: The Agricultural Co-operative Federation of Thailand Printing.

National Economic and Social Development Board. (2017). *The twelfth national economic and social development plan (2017–2021)*. Bangkok, Thailand: Office of the National Economic and Social Development Board.

National Innovation Agency. (2019). *Startup Thailand policy and creative economy development.* Bangkok, Thailand: National Innovation Agency (Public Organization).

OECD. (2018). *The future of education and skills: Education 2030*. Paris, France: OECD Publishing.

OECD. (2019). *PISA 2018 results (Volume I): What students know and can do*. Paris, France: OECD Publishing.

Office of the Education Council. (2020). *National education plan 2017–2036*. Bangkok, Thailand: Prigwhan Graphic.

Office of the National Economic and Social Development Council. (2021). *Master plan under the national strategy. Bangkok,* Thailand: Office of the National Economic and Social Development Council.

Panich, W. (2012). *Ways to create learning for students in the 21st century*. Bangkok, Thailand: Sodsri-Saridwongsa Foundation.

Piaget, J. (1952). *The origins of intelligence in children*. New York: International Universities Press.

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

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

World Economic Forum. (2020). *The future of jobs report 2020*. Geneva, Switzerland: World Economic Forum.