



## UvA-DARE (Digital Academic Repository)

### Exploring embedded computational thinking in STEM teacher education

Pennink, D.; Dekker, I.; Calor, S.; Pijls, M.; Bredeweg, B.

**DOI**

[10.34641/ctestem.2022.454](https://doi.org/10.34641/ctestem.2022.454)

**Publication date**

2022

**Document Version**

Final published version

**Published in**

CTE-STEM 2022 Conference

**License**

CC BY

[Link to publication](#)

**Citation for published version (APA):**

Pennink, D., Dekker, I., Calor, S., Pijls, M., & Bredeweg, B. (2022). Exploring embedded computational thinking in STEM teacher education. In M. Specht, X. Zhang, C. Glahn, & N. Fanchamps (Eds.), *CTE-STEM 2022 Conference* (pp. 90-91). Technische Universiteit Delft. <https://doi.org/10.34641/ctestem.2022.454>

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

# Exploring Embedded Computational Thinking in STEM Teacher Education

Dorrih PENNINK<sup>1\*</sup>, Izaak DEKKER<sup>1,3</sup>, Sharon CALOR<sup>1,2</sup>, Monique PIJLS<sup>1</sup>, Bert BREDEWEG<sup>1,4</sup>

<sup>1</sup>Amsterdam University of Applied Sciences, the Netherlands

<sup>2</sup>Vrije Universiteit Amsterdam

<sup>3</sup>Erasmus University Rotterdam, The Netherlands

<sup>4</sup>University of Amsterdam, The Netherlands

\*[d.h.m.pennink@hva.nl](mailto:d.h.m.pennink@hva.nl), [i.dekker@hva.nl](mailto:i.dekker@hva.nl), [s.m.calor@hva.nl](mailto:s.m.calor@hva.nl), [m.h.j.pijls@hva.nl](mailto:m.h.j.pijls@hva.nl), [b.bredeweg@hva.nl](mailto:b.bredeweg@hva.nl)

## ABSTRACT

Computational thinking (CT) has become a necessity in many professional domains. As such, scholars argue that the acquisition of CT and application should be embedded in existing school subjects. Within the CT literature, a taxonomy distinguishes CT practices in STEM education into four categories: data related, systems thinking, modeling & simulation and computational problem solving (CPSP). Practical applications of these different categories are still limited. This paper presents three examples in which educators of science teachers integrate CT within STEM content knowledge using the above mentioned taxonomy. The first example applies to CPSP and data practices, the second to CPSP exclusively, the final to systems thinking and modeling & simulation. The examples provide practical insight that makes the use of CT in STEM education more tangible for practitioners.

## KEYWORDS

Computational Thinking, Teacher Education, STEM Education, Computational Practice, TPACK

## 1. INTRODUCTION

Digital elements are embedded in every aspect of our society. The Dutch educational system has not yet responded to this development. In the Netherlands, computational thinking (CT) is currently not a formal part of K-12 curricula. While plans are being drawn to include CT in curricula, teacher education needs to equip pre- and in-service teachers with the knowledge and skills to execute these planned innovations.

When classroom activities are becoming computational endeavors, a theoretically grounded operationalization of CT is required that describes the form it should take in STEM classrooms. Weintrop et al. (2016) meet this need, proposing a taxonomy in which four categories of practices are distinguished and illustrated: data related-, systems thinking-, modeling & simulation-, and computational problem solving practices (CPSP).

## 2. EMBEDDED CT

This paper presents three examples in which educators of pre- and in-service science teachers at the Amsterdam University of Applied Sciences (AUAS) integrated CT

within STEM content knowledge using Weintrop et al.'s (2016) taxonomy.

### 2.1. Retrieving information from large datafiles

*The know-it-all has an answer to every question, the wise asks the right questions.* (Bais, 2007)

In a robotics/CT-course, part of the main phase of the bachelor program for pre- and in-service STEM teachers, students were trained to use JavaScript and Python. As a skills training and versatility test, they were asked to re-track information from a COVID-19 dataset (World Health Organization, 2021) and create graphical representations of cases and deaths for different countries in a python notebook environment. Despite limited experience, students proved capable of performing the task (Figure 1). Particularly interesting was the eagerness students demonstrated to find answers to a number of self-formulated questions. An animated societal and ethical discussion followed.

source: <https://covid19.who.int/WHO-COVID-19-global-data.csv> 2021-03-10  
Cumulative nr of cases in USA, UK, NL, G & B per #inhabitants

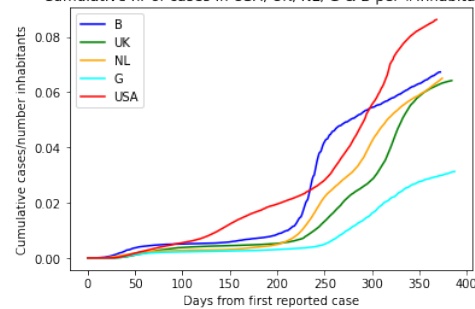


Figure 1. Graphical representation task based on WHO-COVID dataset

Practicing dataset skills with a specific dataset showed an un-foreseen educational gain. Students surpassed the task-given boundaries and became architects of a personally relevant learning process. We regard this as an added value of integrating CT and content knowledge: creating a learning environment in which meaningfulness is shaped by learners themselves using CT-tools to answer their own questions.

### 2.2. Use of a CT-design map

In the same robotics/CT course students were trained to design programs for several tasks, partly self-defined.



©Author(s) 2022. This work is distributed under the Creative Commons Attribution 4.0 License. This license allows anyone to redistribute, mix and adapt, as long as credit is given to the authors.

Although usually capable of performing such tasks, students failed to explicitly identify key concepts of CPSP in their code. To address this problem the CT-design map was introduced in order to disentangle key-concepts. Figure 2 is (part of) a student made example. The top row shows the task description, following rows contain a description of subproblems with associated code lines and an explanation and/or description of action(s) performed. Bottom rows (not in figure) describe testing and debug processes.


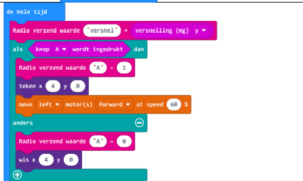
<b>Taakomschrijving</b>	Ontwerp een programma dat dient als afstandsbediening voor de Bit:Bot. De afstandsbediening kan ten alle tijden gebruikt worden, zodat de besturing van de Bit:Bot overgenomen kan worden. <ul style="list-style-type: none"> <li>Als 'A' wordt ingedrukt, dan gaat de Bit:Bot naar rechts</li> <li>Als 'B' wordt ingedrukt, dan gaat de Bit:Bot naar links</li> <li>Als 'A + B' wordt ingedrukt, dan gaat de Bit:Bot rechtdoor</li> </ul>	
<b>Onderdeel</b>	<b>Programmeregels</b>	<b>Toelichting</b>
Radiofrequentie en drempelwaarde juist instellen.		Als de Micro:Bit wordt aangezet, is het nodig om te schakelen naar groep 1 (zoals ook bij alle andere opdrachten het geval was).
Eerste mogelijkheid; Knop 'A' wordt ingedrukt		Als 'A' wordt ingedrukt, dan wordt er een signaal naar de Bit:Bot verzonden waardoor de linker motoren gaan draaien op een snelheid van 60%. Als gevolg hiervan zal de Bit:Bot rechts afslaan. Als 'A' wordt losgelaten, dan zal de Bit:Bot autonoom verder rijden en treedt de lichtsensor weer in werking.

Figure 2. CT-design map – student's work

Using this map has several advantages. Firstly, it stimulates CPSP in a structured manner. Secondly, key facets of CT, such as abstraction, decomposition, pattern recognition and algorithmic thinking are clearly distinguishable. This contributes to the development of consciously competent teachers. An additional merit is offered by the 'explanation' column. In science education we emphasize the value of 'thinking-back-and-forth' between representations. Students are trained in 'talking-science', a type of storytelling to describe phenomena from different perspectives. The articulation of code is an expression of this form of thinking.

### 2.3. Systems thinking approach of modeling

*All models are wrong, some are useful. (Anonymous)*

With the ubiquity of covid-, climate- and other models, the societal relevance of dynamic modeling is indisputable. By embedding modeling in science education students learn about the possibilities and limitations of models. Furthermore, dynamical models are used to enhance conceptual understanding and test hypotheses. A complication in teaching modeling is the high cognitive load undermining learning effects (Van Buuren, Heck, and Ellermeijer, 2016). To cope with this problem a new strategy is explored in a kinematics and dynamics course in physics teacher education at AUAS. Before confronting students with modeling tasks, models are presented as concept maps constructed with modeling software.

The map in Figure 3 represents the relation between two of Newton's laws and kinematic quantities. As in the previous example students are encouraged in thinking-back-

and-forth between the different perspectives (i.e. concept map, graphs, equations, movement description etc.).

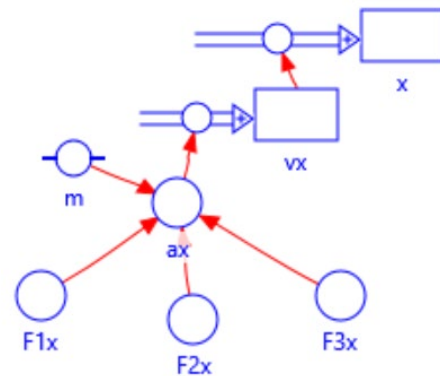


Figure 3. Systems thinking concept map

With this approach we plan to contribute to a better understanding of related concepts in mechanics and increase the learning efficacy of modeling tasks.

## 3. FUTURE WORK

In the near future embedded acquisition and application of CT will be part of existing school subjects. At the AUAS, science teacher education is preparing for this change with the introduction of a CT-curriculum. We explicitly articulate CT knowledge and skills, we investigate further CT enrichments embedded in STEM content knowledge, and we study, develop and apply the pedagogy of a subject-integrated approach. Our intention is not to deliver IT experts, but innovative professionals instead, with sufficient experience and self-confidence to implement the curriculum revision in a meaningful way in their own teaching practice.

The presented examples demonstrate our explorations to develop a CT-content knowledge integrated curriculum.

## 4. REFERENCES

- Bais, S. (2007). *De sublieme eenvoud van relativiteit: een visuele inleiding*. Amsterdam University Press.
- Van Buuren, O., Heck, A. & Ellermeijer, T. Understanding of Relation Structures of Graphical Models by Lower Secondary Students. *Res Sci Educ* **46**, 633–666 (2016). <https://doi.org/10.1007/s11165-015-9474-x>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology*, *25*(1), 127-147.
- World Health Organization. (2021) Coronavirus COVID-19 daily new and cumulative cases and deaths by country. <https://covid19.who.int/WHO-COVID-19-global-data.csv>