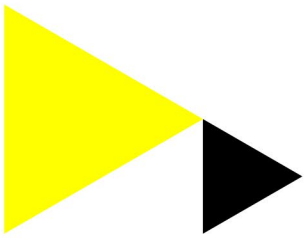
Solving domain-specific problems with computational thinking



 DOI: 10.34641/ctestem.2022.470

Solving Domain-Specific Problems with Computational Thinking

Sharon CALOR[1, 2*], Izaak DEKKER[1, 3], Dorrith PENNINK[1], Bert BREDEWEG[1, 4]
[1]Amsterdam University of Applied Sciences, the Netherlands
[2]Vrije Universiteit Amsterdam
[3]Erasmus University Rotterdam, The Netherlands
[4]University of Amsterdam, The Netherlands

*s.m.calor@hva.nl, i.dekker@hva.nl, d.h.m.pennink@hva.nl, b.bredeweg@hva.nl

ABSTRACT

Computational thinking (CT) skills are crucial for every modern profession in which large amounts of data are processed. In K-12 curricula, CT skills are often taught in separate programming courses. However, without specific instructions, CT skills are not automatically transferred to other domains in the curriculum when they are developed while learning to program in a separate programming course. In modern professions, CT is often applied in the context of a specific domain. Therefore, learning CT skills in other domains, as opposed to computer science, could be of great value. CT and domain-specific subjects can be combined in different ways. In the CT literature, a distinction can be made among CT applications that substitute, augment, modify or redefine the original subject. On the substitute level, CT replaces exercises but CT is not necessary for reaching the learning outcomes. On the redefining level, CT changes the questions that can be posed within the subject, and learning objectives and assessment are integrated. In this short paper, we present examples of how CT and history, mathematics, biology and language subjects can be combined at all four levels. These examples and the framework on which they are based provide a guideline for design-based research on CT and subject integration.

KEYWORDS

Computational thinking, Domain-specific problems, Developed examples, Integration, K-12

1. INTRODUCTION

Computational thinking (CT) was initially introduced by Papert (1980) as a method to perceive relationships between parts of a complex system. Wing (2006) defined CT as a way to solve problems, design systems and explain behavior by exploiting concepts from computer science. Shute et al. (2017) argue that such thinking can in principle also be done without computers, Denning and Tedre (2021) state that CT is in practice intertwined with its application in computers. Concepts from CT have also influenced the way in which we (from the viewpoint of different sciences) explain reality using information processing. Denning and Tedre (2021) therefore propose a twofold definition. On the one hand, CT consists of the ability to design applications that enable computers to perform tasks for us and, on the other hand, of the skills with which we can explain and interpret the world in terms of information processes. Defined in this way, CT is a set of skills essential to every modern profession in which the use of large amounts of data (information) is important. Typical activities associated with this concept in the literature include simulation, data mining, networking, automated data collection, gaming, algorithmic reasoning, robotics, programming, problem solving, modeling, data analysis and interpretation, as well as statistics and probability (Shute et al., 2017). These types of activity require several key skills that are linked in an iterative process: decomposition, abstracting, algorithmic thinking, debugging, iteration, and generalization (Shute et al., 2017).

2. CT-SUBJECT INTEGRATION

In K-12 curricula, CT skills are often taught in separate programming courses. However, literature on transfer of learning (Salomon & Perkins, 1989) suggests that without specific instructions, CT skills will not automatically transfer to other domains in the curriculum when they are developed while learning to program in a separate programming course. Integrating CT in existing courses could be of great added value because CT is often applied in practice in the context of a particular domain.

Yeni et al. (in press) distinguish three phases that are ideally completed when applying CT in a domain based on the process model of Barendsen and Bruggink (2019). In the first phase, a problem is converted into data or processes so that a computer can solve it. A computational solution is then created using an existing or self-developed program. Finally, the computational solution is re-interpreted in the context of the domain.

In addition, Yeni et al. (in press) classify the studies they located in their systematic literature review according to the degree to which CT skills are integrated with the subject-specific problems. At the substitution level, existing programs are applied by the teacher to illustrate a given matter. At the augmentation level, the students can use the programs themselves to find answers to questions without learning how the programs work as part of the lesson. At the modification level, the lesson design is different due to the use of CT: the learning objectives are no longer only focused on subject-related skills but enable students to adapt this subject with the help of CT. At the redefining level, students can use CT to solve questions/problems that cannot be solved without CT, for example, solving a problem by creating algorithms, simulations or programs. Therefore, at the highest level,

domain-specific problems are tackled that can only be solved with CT.

3. EXAMPLES

In this section, we present examples of how history, mathematics, biology and language subjects can be combined with CT at all four levels. These examples and the framework on which they are based provide a guideline for design-based research on CT and subject integration.

3.1. History

Examples of domain-specific problems in history that require CT are questions such as "What role did slavery play within the Dutch East India Company, and how did this role change during the 17th, 18th and 19th centuries?" or "How often were slaves recorded in notarial deeds during the 16th, 17th, and 18th centuries?"

Using Artificial Intelligence the National Archives of the Netherlands have digitally transcribed more than 2 million pages of 17th, 18th and 19th century texts from (among others) the Dutch East India Company and made them publicly available.

On the substitution level, generated data can be used for illustration purposes in the classroom. On the augmented level, students can search the database themselves on the basis of detailed searches and hypotheses to test hypotheses regarding colonial history. On the modification level, students can study if and how the database takes into account how language use changes over time. On the redefinition level, students can formulate and test hypotheses using the database and search strings that consider changes in language use over time.

3.2. Mathematics

An example of a domain-specific problem in mathematics that requires CT is "How to interpret and analyze large datasets?"

Quantifying and visualizing data obtained with the help of statistical software and making statements in the field of explanatory statistics based on such data has CT potential.

On the substitution level, statistical software can be used to illustrate what the median or mode is in a large dataset. On the augmented level, students can process data from large datasets into an appropriate table or graph and test it for value. On the modification level, students can make statements about a population based on sample data and quantify its reliability. On the redefinition level, students can design a plan to obtain answers to a problem statement using large datasets, connect interpretations to the obtained data and interpret the result in terms of the context.

3.3. Biology

An example of a domain-specific problem in biology that requires CT is "How can we, e.g., track and explain biodiversity loss with respect to the bee population?"

The Global Biodiversity Information Facility (http://gbif.org) makes large datasets available that include information on, for example, the diet and reproduction of bees in various European cities.

On the substitution level, the results of studies that employ large datasets are used as examples to support biodiversity theory. On the augmentation level, the teacher formulates questions on the basis of biodiversity theory that students can answer using a dataset. On the modification level, the teacher demonstrates which code can be used to analyze a dataset in Python to answer questions and allows his or her students to practice with this dataset. On the redefinition level, students may modify the code to answer new questions using the available dataset.

3.4. Language

An example of a domain-specific problem in language is "How do you find the most relevant and reliable information for an argument from a nearly infinite dataset of sources?"

Data retrieval is a technique with which data can be efficiently extracted from large datasets. To utilize this potential, it is necessary to formulate search strings with which the search engine can make targeted selections.

On the substitution level, students can use search engines to replace library catalogs (search by author, source, genre). On the augmented level, students can enter and test predefined search strings (search terms that are linked with Boolean operators AND, OR or NOT) in an online database. On the modification level, students can assess which parts of the search string are not functioning properly and require replacing. On the redefinition level, students can formulate, test and modify in iterations a search string that excludes and includes exactly the desired resources from a large dataset.

4. FUTURE RESEARCH

In this short paper, we have presented examples of how history, mathematics, biology and language subjects can be combined with CT on four levels. These examples, and the framework on which they are based, provide a guideline for design-based research on CT and subject integration.

Our ongoing work implements the described examples in the classrooms in which the different subjects are taught.

5. AUTHOR AND CONTRIBUTOR STATEMENT AND DATA ACCESS STATEMENT

5.1. Author and Contributor Statement

5.1.1. Author Statement

Sharon M. Calor: Conceptualization, Writing – Review & Edit

Izaak Dekker: Conceptualization, Writing – Review & Edit

Dorrith H.M. Pennink: Conceptualization, Writing – Review & Edit

Bert Bredeweg: Conceptualization, Writing – Review & Edit

5.2. Data Access Statement

Not applicable.

6. REFERENCES

Barendsen, E., & Bruggink, M. (2019). Het volle potentieel van de computer leren benutten: over informatica en computational thinking. *Van Twaalf tot Achttien*, *29*(10), 16-19. https://onderwijstijdschriftenplein.nl/tplein/van-twaalf-tot-achttien-jrg-29-december-2019-nr-10/

Denning, P. J., & Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, *20*(3), 361-390. https://doi.org/10.15388/infedu.2021.21

Salomon, G., & Perkins, D.N. (1989). Rocky roads to transfer: Rethinking mechanisms of a neglected phenomenon. *Educational Psychologist*, *24*(2), 113–142.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158. https://doi.org/10.1016/j.edurev.2017.09.003

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. http://www.cs.cmu.edu/~wing/

Yeni, S., Grugrina, N., Hermans, F. F. J., Tolboom, J. & Barendsen, E. (in press). Embedding computational thinking in the non-computing subjects: A systematic literature review.