Computational Thinking Skills in Dutch Secondary Education

Natalia Grgorina, University of Groningen, n.grgorina@rug.nl
Erik Barendsen, Radboud University Nijmegen
Bert Zwanenweld, Open University
Wim van de Griff, University of Groningen
Edzard Stoker, Radboud University Nijmegen

Some CT aspects can be recognized in current CS teaching practice. How can we ensure systematic teaching of CT in the CS curriculum?

We shall study the following issues:

1. What is an operational definition of Computational Thinking, tailored to the specific situation and needs of secondary education in the Netherlands?
2. How can students’ CT problem solving skills be assessed?
3. What is a suitable pedagogical approach to teach students and stimulate their learning of CT problem solving skills?

Computational thinking - a new concept

“Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are effectively carried out by an information processing agent.”

A typical CS assignment in secondary education

Make a model / simulation / program for:

- Traffic lights for a busy traffic crossing
- Elevator in an apartment building

With this draft definition we shall establish CS teachers’ PCK on CT through structured interviews (CoRe).

An instrument to assess students’ CT will be developed in the second phase.
A pedagogical approach will be developed in the third phase.
The effects of the curriculum intervention will be assessed in the fourth phase.

Category | Subcategory
--- | ---
Data Collection | Collecting data
 | Selecting relevant data
Data Analysis | Drawing conclusions
 | Finding patterns
 | Making sense of data
Data Representation | Arrange data for analysis
 | Organize/represent data
Problem decomposition | Breaking down tasks
 | Merging subtasks
Abstraction | Finding characteristics
 | Creating models
Algorithms & procedures | Making sequential steps in a specific order
 | Understanding and changing algorithms
 | Making decisions in algorithms
 | Implementing algorithms
Automation | Recognizing different forms of automation
 | Recognizing the advantages of automation
Simulation | Creating pseudo-code
 | Creating models of processes
 | Experimenting
Parallelization | Combine/merge activities

Result of the first phase: final operational definition of CT tailored to the needs of CS course in Dutch secondary education.