Computational Thinking in Dutch Secondary Education
Grgurina, N.; Barendsen, E.; Zwaneveld, B.; Veen, K. van

Published in:
Informatik und Natur: 6. Münsteraner Workshop zur Schulinformatik

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2014

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):

Copyright
Other than for strictly personal use, 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), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.
In 2006, J.M. Wing introduced the term Computational Thinking (CT) [Wi06] indicating a set of analytical skills needed to employ IT techniques and tools effectively. As a part of the first phase of a larger research project where we investigate how to incorporate teaching CT into informatics course in the upper grades of secondary education in the Netherlands [Gr13], we conduct a study aimed at characterizing the present teaching practice around CT, including the pedagogical content knowledge (PCK) [Sh86] of informatics teachers together with their perceptions, beliefs and hindrances.

As a first step towards exploring these perceptions, beliefs and hindrances we analyze the data acquired in an questionnaire on informatics education. In November 2013, the Netherlands Institute for Curriculum development¹ held an online survey among informatics teachers, aimed at determining their enacted informatics curriculum and exploring their ideas about desired changes of it. Upon our request, a question on CT was added to the survey. We asked, “Do you think that CT gets enough attention in your current teaching practice, as far as it concerns the thought process and as far as it concerns the skills? If you answer in no, then what in your opinion causes the lack of it in your teaching practice?” We supplied a short description of CT: “Computational Thinking is a thought process in which one recognizes situations where data

¹ www.slo.nl
organization, data processing and data analysis can be employed effectively and at the same time possess the skills to formulate problems using IT concepts and solve these problems using IT techniques and tools.” 178 teachers out of an estimated population of about 300 filled in the survey. Out of these 178 teachers, 79 answered ‘no’ to this question and proceeded to describe their reasons.

We made a qualitative analysis of their answers through several iterations. Originally we set out with four categories Magnusson uses to describe teachers’ PCK about a particular topic to be taught: (1) goals and objectives, (2) students’ understanding, (3) instructional strategy; and (4) assessment [MKB99]. In subsequent iterations we added new categories concerning external circumstances and conditions, curriculum aspects, policy, recommendations, teachers themselves and to our surprise, a rather large category indicating an alternative understanding of what CT is in the eyes of the teachers.

Our findings indicate that among the teachers who do not believe they pay enough attention to CT in their teaching practice, many believe the causes lie in the curriculum which does not prescribe teaching CT explicitly, inadequate teaching materials or their own inability to teach it effectively. The most interesting findings, however, are those indicating teachers’ understanding of CT is not in line with the ideas of CSTA which suggest that CT can be taught to all students in K-12 at an appropriate level when embedded in an appropriate context [CSTA11]. Some of the teachers, for example, find CT to be too abstract and theoretical for 15-year old students, that there is not enough time during informatics lessons to teach CT, that there is no need to teach CT or that it is difficult to teach it because the IT policy of the school does not emphasize it.

After this first explorative pilot study, the next step in our research will be to interview a number of teachers to establish their PCK about CT problem solving skills through semi-structured interviews based on the Loughran’s Content Representation (CoRe) [LMB04]. Since a number of teachers hold alternative views on the meaning of the notion of CT, we intend to supply them with an interpretation of CT based on the nine categories described by CSTA [CSTA11] and a refinement into 21 subcategories derived by Barendsen and Stoker through a preliminary analysis of teaching materials, see Table 1 [BS13].

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection</td>
<td>Collecting data</td>
</tr>
<tr>
<td></td>
<td>Selecting relevant data</td>
</tr>
<tr>
<td>Data Analysis</td>
<td>Drawing conclusions</td>
</tr>
<tr>
<td></td>
<td>Finding patterns</td>
</tr>
<tr>
<td></td>
<td>Making sense of data</td>
</tr>
<tr>
<td>Data Representation</td>
<td>Arrange data for analysis</td>
</tr>
<tr>
<td></td>
<td>Organize/represent data</td>
</tr>
<tr>
<td>Problem decomposition</td>
<td>Breaking down tasks</td>
</tr>
<tr>
<td></td>
<td>Merging subtasks</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Finding characteristics</td>
</tr>
<tr>
<td></td>
<td>Creating models</td>
</tr>
</tbody>
</table>

Since a number of teachers hold alternative views on the meaning of the notion of CT, we intend to supply them with an interpretation of CT based on the nine categories described by CSTA [CSTA11] and a refinement into 21 subcategories derived by Barendsen and Stoker through a preliminary analysis of teaching materials, see Table 1 [BS13].
Table 1: Categories and subcategories of Computational Thinking

| Algorithms & procedures | Making sequential steps in a specific order  
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 |

We intend to focus only on those (sub)categories which are directly related to problem solving. This study will yield a characterization of PCK of informatics teachers in Dutch secondary education concerning CT problem solving skills.

References


