Assessment of Modeling and Simulation in Secondary Computing Science Education

Natasa Grgurina
University of Groningen
Groningen, The Netherlands
n.grgurina@rug.nl

Erik Barendsen
Radboud University
Nijmegen, The Netherlands
Open Universiteit Nederland
Heerlen, The Netherlands

Cor Suhre
University of Groningen
Groningen, The Netherlands

Bert Zwaneveld
Open Universiteit Nederland
Heerlen, The Netherlands

Klaas van Veen
University of Groningen
Groningen, The Netherlands

ABSTRACT

The introduction of the new computing science curriculum in the Netherlands in 2019 raises the need for new evidence-based teaching materials that include practical assignments and guidelines for their assessment. As a part of our research project on teaching Computational Science (modeling and simulation), we participate in these efforts and developed a curriculum intervention including a practical assignment and an accompanying assessment instrument consisting of grading rubrics based on the SOLO taxonomy. In this research paper we focus on the assessment instrument. We describe its development and report on a pilot study carried out in the secondary computing science course implementing the curriculum intervention. The instrument proved to be reliable and effective in tracing high and low levels of the students' achievements in modeling and simulation projects and exposed the expected differences in performance levels of various groups of students, which renders it useful for both formative and summative assessment. Furthermore, our application of the instrument has provided new insights into the needs of specific groups of students to receive instruction prior to and during the work on the assignments.

CCS CONCEPTS

• **Social and professional topics → Computer science education**; *Computational science and engineering education*; *K-12 education*;

KEYWORDS

Modeling and simulation, Assessment, SOLO taxonomy, Secondary computing education

WiPSCE '18, October 4–6, 2018, Potsdam, Germany

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/3265757.3265764

ACM Reference Format:
Natasa Grgurina, Erik Barendsen, Cor Suhre, Bert Zwaneveld, and Klaas van Veen. 2018. Assessment of Modeling and Simulation in Secondary Computing Science Education. In *Proceedings of the 13th Workshop in Primary and Secondary Computing Education (WiPSCE '18), October 4–6, 2018, Potsdam, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3265757.3265764

1 INTRODUCTION

In the Netherlands, computing science (CS) is an elective subject in grades 10 and 11 of the senior general secondary education spanning grades 7 through 11 (in Dutch: HAVO) and in grades 10 through 12 of the pre-university education spanning grades 7 through 12 (in Dutch: VWO). The new 2019 secondary CS curriculum recognizes the importance of modeling and includes an elective theme comprised of modeling and simulation, together called Computational Science. It is described by the high-level learning objectives: "*Modeling: The candidate is able to model aspects of another scientific discipline in computational terms*" and "*Simulation: The candidate is able to construct models and simulations, and use these for the research of phenomena in that other science field.*"[2]. The curriculum does not provide further details about these objectives, instruction or assessment. In line with the Dutch tradition, this is left to educators and authors of teaching materials. The elaboration of these learning objectives, the development of teaching materials, assessment tools and teacher training courses are already taking place. We participate in these endeavors with practical assistance to teachers developing teaching practices that help to attain these objectives, and by monitoring these developments through research.

This study is a part of a larger research project on teaching Computational Science in the context of CS in Dutch secondary education, investigating pedagogical aspects as well as teachers' pedagogical content knowledge (PCK) about modeling. (For clarity, in this paper the terms modeling, simulation modeling and computational science all refer to the learning objective computational science.) Following Magnusson et al. [23], we distinguish four elements of content-specific pedagogy: (M1) goals and objectives, (M2) students' understanding and difficulties, (M3) instructional strategies, and (M4) assessment. Previously, we refined the CSTA definition of computational thinking (CT) [14], made initial explorations of teachers' PCK [15, 16] and of the computational modeling process [13], obtained an operational description of the intended

learning outcomes (ILO) of the learning objective Computational science — thus focusing on Magnusson's element M1, observed students working on modeling tasks — focusing on Magnusson's element M2, established what data sources were suitable for assessment — Magnusson's element M4 [17], investigated teachers' initial pedagogical content knowledge on modeling and simulation [11], and finally, proposed an assessment instrument [12] — Magnusson's element M4.

In this study, we focus on monitoring the levels of understanding in the learning outcomes of students engaging in modeling projects — Magnusson's element M4. We aim to examine the agreement and validity of an instrument to assess students proficiency in modeling a (problematic) situation and to provide answers to improve the situation. We seek answer to these questions:

(1) Can the instrument be used by different teachers without having a distinguishable effect on the assessment?
(2) Does the instrument allow for a valid measurement of students' proficiency level?

Two steps are taken to obtain the necessary data to answer both questions. First, inter-rater agreement is assessed by having two teachers rate the same products and compute the inter-rater agreement. Second, the proficiency levels of projects made by students of different education level — HAVO and VWO — are compared to investigate whether differences are in the expected direction.

2 BACKGROUND AND RELATED WORK

2.1 Computational Thinking: Modeling

Formulating problems in a way that enables us to use a computer to solve them and representing data through abstractions such as models and simulations are integral parts of computational thinking (CT) [9]. With the arrival of computers into schools, new venues are created to aid students' learning in various disciplines through the use of computer models [5, 27]. Wilensky argues, "*Computational modeling has the potential to give students means of expressing and testing explanations of phenomena both in the natural and social worlds*"[33], as do Nowack and Caspersen [7]. Indeed, modeling plays a significant role in the development and learning of science [20] and CS equips the students to actively engage in learning science by providing tools and techniques to engage in modeling, thus enabling them to provide meaning to the learning both of the discipline at hand [10] and CS.

In the new 2019 CS curriculum, for the intended learning outcomes of the learning objective Computational science, in one of our previous studies we developed an operational description that describes the modeling cycle for simulation modeling through its elements purpose, research, abstraction, formulation, requirements/specification, implementation, verification/validation, experiment, analysis, and reflection. Furthermore, in that study we advocated to use agent-based modeling (ABM) when teaching Computational science since it is a suitable simulation modeling method for use in secondary CS education [17] and here we focus specifically on ABM.

2.2 Documenting models

In order to describe what is the purpose of a model, how does it work and other relevant details, it is necessary to document the model. Several techniques have been proposed to do this in order to help to understand a model, to facilitate completeness of the description, and to make it easier to reproduce a model.

The ODD protocol is specifically devised to standardize the descriptions of individual-based and agent-based models [18] [19]. It describes a model in terms of its overview, design concepts and details — hence the acronym ODD. In the updated ODD protocol, the overview contains the elements (1) purpose, (2) entities, state variables and scales, and (3) process overview and scheduling. The eleven elements of the design concepts are the basic principles, emergence, adaptation, objectives, learning, prediction, sensing, interaction, stochasticity, collectives and observation. Finally, the details deal with the elements initialization, input data and submodels. However, this approach to documenting models has several weaknesses: due its textual nature, it is inherently ambiguous and furthermore, it does not allow for documentation of all the relevant details and thus hampers the reproduction of models, as noted by Amouroux et.al. They charted the strengths and weaknesses of the ODD protocol and suggest the addition of an "*Execution environment*" section to aid the model replication [1]. A different approach to documenting models recognizes the common traits of agent-based modeling and object-oriented programming and suggests to expand the Unified Modeling Language (UML) to accommodate the specifics of ABM. The UML supports the following kinds of models: static models, dynamic models, use cases, implementation models and object constant language (OCL) [28]. Odell et al. suggest the agent-based extension AUML, that is, "*agent-based extensions to the following UML representations: packages, templates, sequence diagrams, collaboration diagrams, activity diagrams, and statecharts.*"[26]. Similarly, Bauer et al. propose Agent UML with four agent-based extensions to UML representations: packages, templates, sequence diagrams and class diagrams [3]. Muller et al. go a step further to explore the suitability of particular types of model descriptions for specific intended purposes. To this end they distinguish eight possible purposes of models: communication of the model — to peers, for education or for stakeholders; in-depth model comprehension, model assessment — to establish its suitability for its purpose, model development — design and collaborative, model replication, model comparison, theory building, and finally, code generation. They go on to assess how well these purposes are met by the different description types: natural language — either with a prescriptive structure, such as ODD protocol, or without it, such as verbal description; formal languages — ranging from various ontologies, source code, pseudo code to mathematical description, and finally graphics — either formal such as UML, or non-formal. In case of communication for education, they suggest that non-formal verbal description, source code made with the program-level tools (such as, for example, NetLogo [32]) and non-formal graphics are among the most suitable description types, while formal descriptions with ODD protocol or UML as well as non-specialized programming languages are less suitable. They conclude by suggesting "*a minimum standard of model description*

for good modelling practice, namely the provision of source code and an accessible natural language description."[25]

2.3 Assessment

Brennan and Resnick focused on assessment of the development of CT during learning in informal settings and developed a CT framework distinguishing three dimensions: computational concepts describing the concepts designers employ when programming, namely *sequences, loops, parallelism, events, conditionals, operators, and data*; computational practices describing the practices designers employ when engaging with the concepts, namely *being incremental and iterative, testing and debugging, reusing and remixing, and abstracting and modularizing*, and computational perspectives describing the perspectives designers form about the world around them and about themselves, namely *expressing, connecting and questioning*. [6] Zhong et al. brought these three dimensions of CT into the classroom when designing an assessment framework for elementary school students and they redefined them as follows: computational concepts as "*objects, instructions, sequences, loops, parallelism, events, conditionals, operators, and data*"; computational practices as "*planning and designing, abstracting and modeling, modularizing and reusing, iterative and optimizing, and testing and debugging*", and computational perspectives as "*creative and expressing, communicating and collaborating, and understanding and questioning*"[36]. Using this framework, Lye and Koh analyzed 27 intervention studies in K-12 aiming at the development of computational thinking through programming and found that the majority focuses on computational concepts and only six on computational practices. In order to promote focus on computational practices and computational perspectives in a K-12 classroom, they suggest an instructional approach providing "*a constructionism-based problem-solving learning environment, with information processing, scaffolding and reflection activities.*"[22] Brennan and Resnick offer six suggestions for assessing computational thinking via programming, among others to make assessment useful to learners, to incorporate creating and examining artifacts, and to have the designer illuminate the whole process. [6]. These views are corroborated by the findings in our prior study on CS teachers' pedagogical content knowledge (PCK) of modeling and simulation, where we learned that the interviewed teachers mostly suggest a hands-on approach to learning and that the preferred assessment form for most of them would be a practical assignment lasting several weeks, where student groups would construct models and use them to run simulations and conduct research while extensively documenting the whole process. At the same time, we observed a great diversity in the assessment criteria teachers mentioned, but very few corresponding quality indicators used to judge to what extent these criteria are met [11].

In the eyes of the students, the assessment defines the actual curriculum, according to Biggs and Tang. In their constructive alignment network, the curriculum is stated in the form of clear intended learning objectives (ILO) specifying the required level of understanding, the teaching methods engage students in doing things nominated by the ILO's and the assessment tasks address these ILO's. Learning outcomes can be classified using the Structure of the Observed Learning Outcome (SOLO) which describes the learning progress through five levels of understanding. The first three levels — prestructural, unistructural and multistructural — are considered to be quantitative in the sense that prestructural indicates missing the point, unistructural means meeting only a part of the task and multistructural shows a further quantitative increase in what is grasped: "*knowing more*". Relational, on the other hand, indicates a qualitative change indicating conceptual restructuring of the components — "*deepening understanding*", and extended abstract takes the argument into a new dimension [4].

Looking into the use of SOLO taxonomy to assess the novice programmers' solutions of code writing problems, Whalley et al. noted that previous research had indicated difficulties in mapping from student code to the SOLO taxonomy "*since the mapping process seems very context bound and question specific*"[30]. Indeed, Meerbaum-Salant et al. remarked that while the strength of the SOLO taxonomy lies in the fact that it offers a holistic, rather than a local perspective, "*using [it] for various types of activities, simultaneously, is not straightforward*"[24]. When they set out to combine the Revised Bloom taxonomy [21] with SOLO in order to construct assessment for programming tasks of novice programmers, they started out with the interpretation of SOLO as five ordered categories:

Prestructural Mentioning or using unconnected and unorganized bits of information which make no sense.

Unistructural A local perspective — mainly one item or aspect is used or emphasized. Others are missed, and no significant connections are made.

Multistructural A multi-point perspective — several relevant items or aspects are used or acknowledged, but significant connections are missed and a whole picture is not yet formed.

Relational A holistic perspective — meta-connections are grasped. The significance of parts with respect to the whole is demonstrated and appreciated.

Extended abstract Generalization and transfer — the context is seen as one instance of a general case [24]

The issue of assessing the learning of the students engaged in larger programming projects attracts attention as well. Casto and Fisler explored how to track program design skills through the entire CS1 course and suggest a multi-strand SOLO taxonomy, thus corroborating the idea that using SOLO taxonomy simultaneously for various types of activities is not straightforward. They suggest a multi-strand SOLO-taxonomy without the extended abstract level since none of the students in their study reached that level [8]. A multi-strand SOLO taxonomy is in line with the idea that one assessment task might address several ILOs and vice versa, one ILO might be addressed by several assessment tasks [4]. Assignments for complex tasks encompassing diverse ILOs — such as in case of Computational science, thus going through a modeling cycle by formulating a problem, pinpointing the research question, designing a model and using it to answer the research question — warrant the elaboration of criteria defining performance for each of the ILOs involved.

3 ASSESSMENT INSTRUMENT

Based on these findings, we developed constructionist teaching material about agent-based modeling with NetLogo meant for the CS students in the 11^{th} and 12^{th} grades of both HAVO and VWO who

are preferably no novice programmers but rather somewhat experienced, probably in other programming languages. The teaching material covers all the aspects of the ILO's of Computational science we identified earlier [16], and addresses not only computational concepts such as programming to implement the model, but also computational practices such as the validation of the model and computational perspectives such as formulating the research question to be answered through the use of the model. Together with this teaching material, we also developed an assessment instrument on which we focus here.

Following the teachers' suggestions about desirable form of assessment [11] and our findings about suitability of various data sources for assessment confirming the suitability of project documentation [17], we developed an assessment instrument consisting of a practical assignment where students design models and use them to conduct research of phenomena in another science field, and accompanying rubrics for assessment based upon the project documentation and models themselves (i.e. program code). Guided by the suggestions for the rubrics construction by Wolf and Stevens [35], from the modeling cycle we first identified the criteria that defined performance as: stating the case and the research question, designing the model and implementing it, validation, experiment, analysis, answering the research question, and reflection. Subsequently, we designed a practical assignment that provides several cases and research questions for students to choose from, a detailed description of the modeling process they need to engage in, and a corresponding rubrics based on SOLO taxonomy.

An example of the cases provided is the question whether sustainable human life is possible on Mars. The students are pointed to the websites of NASA and SpaceX to learn about the current state of affairs and subsequently have to explore whether, after the initial supplies and shelter were delivered, it would be possible to produce sufficient water, air and food to survive and thus whether it would be possible to found a sustainable human colony on Mars. Among other cases are the questions, what is better for traffic flow on a junction: a roundabout or traffic lights, and to investigate the optimal number and task division of bank counters as to minimize the waiting time of the customers with various needs. In line with our dedication to stimulate student engagement, the students are allowed to come up with their own research questions instead.

While these assignments allow students to make their own choices and decisions when designing their models, we needed a standard that allows educators using our assessment instrument to easily assess the quality of their students' models. To set such a standard, for each case we constructed a *minimal expert model* — a description of a minimal model that fulfills the stated purpose and contains all the necessary agents with their correct behavior and interactions. Since we wanted these minimal expert models to be described on a conceptual level only, we refrained from implementing them in NetLogo because we believed that that would hinder the assessment process rather than contribute to it. Instead, for the models that our students are expected to make — two-dimensional, containing only a few types of agents, no links and no advanced behavior such as learning or sensing — we devised our own format to describe them. This format is partly narrative, borrows aspects of class diagrams from UML and exploits the idea of graphical representation of timed automata where it is possible to require that particular state transitions are allowed only under certain conditions, or only synchronously with other state transitions [29]. Here we illustrate this approach with the description of our minimal expert model of the roundabout. First of all, there is the agent type[1] *vehicle* with its representation — inspired by UML class diagram — stating that an agent of this type has properties[2] *current position* and *target position*, and behavior consisting of actions *show up*, *wait*, *move* and *leave*.

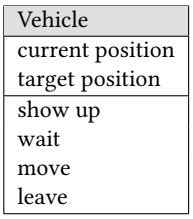

Figure 1: UML class diagram for *vehicle*

Then there is a graphical representation of a state diagram of a *vehicle* which is interpreted as follows: during the NetLogo *setup* procedure, the first transition from the state *begin* to the state *ready* occurs, and that is the action *show up*. The NetLogo *go* procedure then runs repeatedly until the program stops and according to our convention, every time it runs, exactly one transitions originating from the *ready* state takes place, and when the procedure has finished one run the *vehicle* is back in the *ready* state unless it reached the *gone* state. This is not to say that during one run of the *go* procedure an agent may engage in one state transition only: rather, it means that one full cycle of actions emanating from the *ready* state takes place. For each transition, there might be conditions that need to be met in order for the transition to occur, and properties could be updated. For example, *move* only occurs if the position in front of the *vehicle* is free and then its property *current position* is updated. State transitions could be synchronized with each other as well, such as in the famous Wolf Sheep Predation model: a *wolf* can *eat* only if one of the *sheep* simultaneously dies [31].

Finally, there is an additional textual description of a number of relevant aspect of the model. It mentions that a roundabout itself can be modelled as well while that is not strictly necessary since the *vehicles* know where they are through their *current position* property; that the *show up* action does not necessarily require a *vehicle* to stop in front of an empty roundabout, and that it is up to the modeler to decide on design details such as how far can a *vehicle move* during one run of the *go* procedure.

As elaborated in the section on assignment, the evidence the students provide about their models is twofold: the textual descriptions they formulate when designing the model and the implemented model, i.e. NetLogo program code [25]. Our description format helps educators to distil the relevant aspects of the models from the descriptions and code students turn in and to employ the rubrics based on the SOLO categories.

[1] NetLogo speaks of breeds of agents, but we use the term type to cater for those not familiar with NetLogo.
[2] NetLogo speaks of agent's own variables

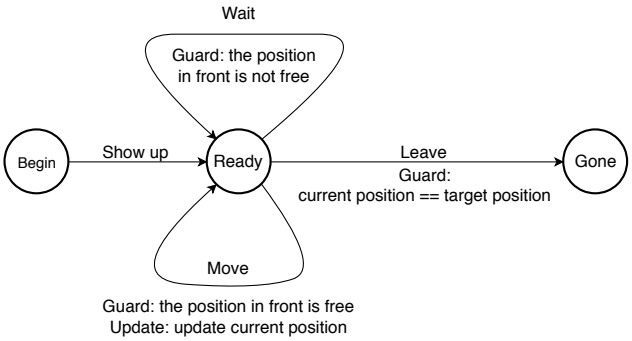

Figure 2: State diagram for *vehicle*

With our format in mind, for the characterization of these SOLO categories we followed the local-to-global perspective. The first three levels describe quantitative progression. Prestructural indicates not answering the question at all or missing the point. Unistructural indicates fragmentary knowledge from a local perspective like mentioning only some of the relevant aspects — such as only a few agents or only some of their states — or missing important details. Multistructural indicates a more complete and coherent multi-point picture of the aspect under scrutiny — like listing all necessary agents and their states — but missing significant connections and without substantiating, clarifying, analyzing or explaining. These activities, however, are characteristic for the last two levels that add qualitative aspects: the relational level requires additionally the understanding of the relations among the parts of the aspect under scrutiny, such as recognizing all the actions an agent can perform, properly identifying conditions for these actions to occur and acknowledging their consequences. Finally, generalizing or theorizing — what if? — about aspects indicate going beyond what was given and typify the extended abstract level [4, 24]. In chapter 3.2, we elaborate the description of the SOLO levels for each criterium defining performance in detail.

3.1 Assignment

The assignment consists of a number of questions the students need to answer in writing while designing their model and using it to answer their research question, and of course, the task of implementing the model. After forming groups and choosing a case to model, the students answer the following questions:

Case and research question. Describe what you are going to model and with what purpose: (1) What do you know about this phenomenon? If need be, carry out the necessary research. (2) What part of your phenomenon would you like to build a model of? (3) What do you hope to observe from this model? (Questions 2 and 3 suggested by [34].)

Design the model. Design and describe the model following the questions listed here. Report the considerations and choices you make. (E.g., "*The sheep can reproduce. If two sheep meet, there is a chance of 20% that a new sheep will be breed. We decided not to take into account the gender of the sheep because that is not relevant in this case.*") (1) What are the principal types of agents involved in this phenomenon? (2) In what kind of environment do these agents operate? Are there environmental agents? (3) What properties do these agents have (describe by agent type)? (4) What actions (or behaviors) can these agents take (describe by agent type)? (5) How do these agents interact with this environment or each other? (6) If you had to define the phenomenon as discrete time steps, what events would occur in each time step, and in what order? (All questions suggested by [34].)

Implement the model. Implement the model in NetLogo. Write your code in small chunks and keep testing!

Validate the model. (1) Microvalidation: to what extent does the agents' behavior resemble the observations of the phenomenon in reality? If the behaviors are (somewhat) dissimilar, is this variation relevant to your research question? (2) Macrovalidation: to what extent does the behavior of the system as a whole resemble the observations of the phenomenon in reality? If the behavior is (somewhat) dissimilar, is this variation relevant to your research question?

Experiment, analysis and conclusion. Use the model to answer your research question: (1) Describe the experiment in detail. If you use Behavior Space, report the number of experiments conducted and the parameters used. (2) Report the findings in an appropriate manner (e.g., a narrative, a table, a graph, etc.) (3) Analyze the results. (4) Answer the research question.

Reflection. Reflect on your modeling process: (1) What went well and what could be better? (2) Did you make any assumptions which, in retrospect, you would like to reconsider? (3) Are there any aspects of your model which you would like to change? Are there any aspects of your model (agents or behavior) you decided not to include in you model while now you believe they do need to be included? Make a wish list of aspect of your model that need to be added, removed or changed in the next version of the model.

Students were also asked to log their activities, problems, and successes; possible explanations for problems and successes, and, lessons learned.

3.2 Grading Rubrics

After we identified the criteria that defined performance, we created performance descriptions [35] to describe the appropriate level of understanding for intended learning outcomes [4]. Here we quote these descriptions:

Case and research question.

(1) What do you know about this phenomenon? If need be, carry out the necessary research.
   - **Prestructural** Nothing or simplistic idea of the phenomenon. Performed no research.
   - **Unistructural** Some general description. Performed no research or only limited to isolated aspects of the phenomenon
   - **Multistructural** Performed some research. Able to name more relevant aspects of the phenomenon, but mentions no relations among these aspects

Relational Performed research. Complete idea of the phenomenon. Able to name relevant aspects of the phenomenon, have insight into relations among these aspects

Extended abstract Additionally, described the relation of this phenomenon to other phenomena in the world and/or conceptualized this phenomenon so as to be able to use it other contexts restricted and its relevance explained. Stated its relevance for other phenomena

(2) What part of your phenomenon would you like to build a model of?

Prestructural Nothing, or a few non-specific remarks but missing the point

Unistructural Few isolated aspects of the phenomenon identified

Multistructural Described what (part of the) phenomenon is being modeled.

Relational Clearly explained what (part of the) phenomenon is being modeled, together with limits of what is being modeled and its significance for the whole.

Extended abstract Additionally, theorize about possible generalization of the model or transfer into a different context.

(3) What do you hope to observe from this model?

Prestructural Research question not clear

Unistructural Identified the question from a local perspective

Multistructural Described the question from a multi-point perspective

Relational The research question clear and predicts possible outcomes.

Extended abstract Additionally, theorize about possible generalization or transfer into a different context.

Design the model and implement it.

Prestructural No agents mentioned

Unistructural A few agents and actions identified

Multistructural Several agents and actions described.

Relational Agents, actions and interactions correct and substantiated. Their contribution to the whole acknowledged.

Extended abstract Additionally, generalize or hypothesize about similar models in different contexts or extend the model beyond the minimal requirements.

Validate the model.

Prestructural Nothing. No working program.

Unistructural Identified some resemblances and differences between the model and reality. Relevance for the research question not clear.

Multistructural Described resemblances and differences between the model and reality. Relevance of the differences for the research question not clear

Relational Resemblances and differences between the model and reality described. Analyzed and explained their relevance for the research question.

Extended abstract Additionally, hypothesized over model adjustments to improve its validity for a more general purpose

Experiment.

Prestructural nothing

Unistructural a few model runs (i.e. simulations) without a clear plan

Multistructural Simulations performed systematically but the relevance for the research question not clear (e.g. not clear why certain data is gathered)

Relational Simulations performed systematically. The relevance for the research question is made clear.

Extended abstract Additionally: The relevance for the research question is clear and substantiated.

Analysis.

Prestructural nothing

Unistructural some results reported

Multistructural results described in an appropriate manner

Relational results described in an appropriate manner. The relation between the values of model parameters and the output data analyzed

Extended abstract Additionally, explain or hypothesize about the relation between the values of model parameters and the output data

Answer the research question.

Prestructural No answer

Unistructural Simple answer

Multistructural Elaborate answer, but the coherent picture not formed

Relational Elaborate answer, coherent picture of the parts and the whole formed

Extended abstract Additionally, discussion

Reflection.

Prestructural No answer

Unistructural Few aspects mentioned

Multistructural Several aspects described

Relational Aspects analyzed and explained

Extended abstract Additionally, discuss the possible consequences in the future

4 METHOD

4.1 Educational context

Four classes participated in this study: one 11^{th} grade VWO class and two 12th grade VWO classes which we all treated as one for the purpose of this research, and one 11^{th} grade HAVO class. The data were collected in two schools in classes that were taught by two teachers: the first author and her colleague who worked in both schools. All the students have previously learned to program in Python or a similar high-level textual programming language. The course on Computational Science lasted eight weeks in total. The first five weeks were dedicated to instruction using the teaching material we developed for our curriculum intervention, and during the last three weeks, the students formed groups of two or three (a few students choose to work alone) and worked on the practical assignment. In total there were eight groups in 11^{th} grade HAVO, five in 11^{th} grade VWO and twelve in 12^{th} grade VWO. After choosing the cases to model, the students went on to answer the questions form the assignment and to develop their models in NetLogo. The

students from the two 12th grade VWO classes presented their models in the classroom and got feedback from other students. In other classes there was no opportunity to organize presentations. Finally, the students answered the last questions from the assignment and turned in their documentation and models, i.e. Netlogo code.

4.2 Data collection

Both the teachers assessed the students' work — project documentation and program code — using the rubrics presented here, assigning 0 up to 4 points for the prestructural up to extended abstract level, respectively, First, they separately assessed work of two groups. They compared their scores, and then the scoring guidelines and the interpretation of the rubrics were fine-tuned where necessary. Additionally, they agreed to take into account the answers students supplied while answering other questions — e.g., elaborating on the research of the phenomenon under scrutiny while answering the question about validation. Then, one teacher assessed the work of all the groups while the other teacher assessed only the work of the 12th grade VWO groups in order to establish the inter-rater agreement.

4.3 Analysis

Our aim was to investigate the inter-rater agreement and the discriminative validity of instrument. The inter-rater agreement was evaluated by computing Krippendorf's alpha for each of the criteria. Discriminative validity was assessed by comparing the scores of HAVO students and VWO students. While it is to be expected that both HAVO and VWO students can be taught to design and implement a model equally well, VWO are expected to outperform HAVO students in defining and analyzing the consequences of manipulation of factors in problem contexts due to their ample preparation in scientific thinking. One-tailed t-tests for independent samples were used to evaluate these expectations. We expect that the majority of scores vary between score 0 (prestructural) and 3 (relational). If the projects of VWO students are more often awarded a score of 4 (extended abstract) than those of HAVO students, this too can be regarded as (partial) evidence that the instrument differentiates satisfactorily between students' level of learning outcomes. In order to see whether the students' results meet our expectations and in search of possible explanation for the observed differences in the performance between HAVO and VWO groups, we additionally performed qualitative in-depth analysis of the students' projects.

5 RESULTS

All the results of the assessment using the grading rubrics are visually presented in Figure 3 and Figure 4, and in aggregate form as mean values of scores per school type (HAVO or VWO) per criterium in Table 1, where the significance levels of the t-tests are presented too. In the figures, each row represents the results of one student group for the nine criteria assessed, and the height of each block represents the score, ranging from 0 for prestructural level to 4 for extended abstract level.

To assess the inter-rater agreement, we computed Krippendorf's alpha assuming ordinal scale from the scores of the twelve 12th grade VWO project assessed by both teachers. The result is 0.78 which is a satisfactory value. Then, we used t-tests to find out

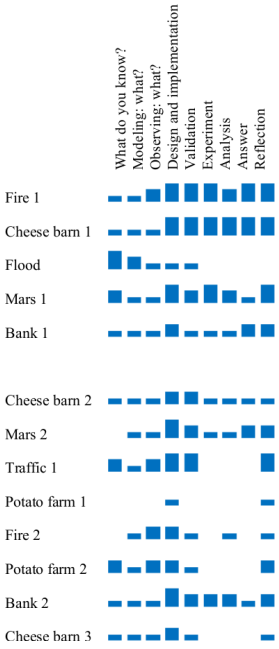

Figure 3: Cases and scores of the HAVO groups

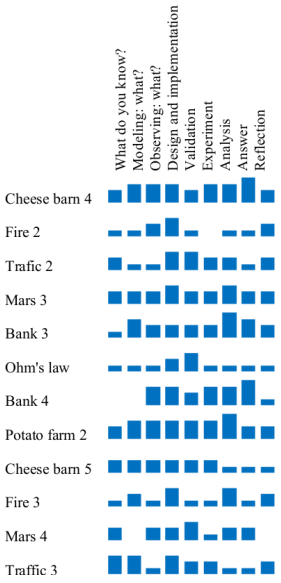

Figure 4: Cases and scores of the VWO groups

whether the achievements of all the VWO groups taken together differ significantly from the achievements of the HAVO groups. We used a one-tailed t-test with significance level of 0.05. The results show that the groups differ significantly (printed bold in Table 1) for all the criteria except designing and implementing model, and for reflection.

We now examine the students' projects per criterium and per type of school, i.e. HAVO vs. VWO. We state our findings in general terms and illustrate them with characteristic text segments taken from the data.

Case and research question. While many groups — in HAVO as well as in VWO — did not engage in any form of research and relied on their existing knowledge of the phenomenon under scrutiny, we see significant differences between HAVO groups on one hand and VWO groups on the other. Among the HAVO groups, none rose above the quantitative levels of SOLO. For example, the group exploring life on Mars (case Mars 2), when answering what do they know about it, simply wrote, "*that life on Mars will not happen for a long time*". However, this group made a good model and showed reasonable performance in the rest of the assignment. In the VWO groups, we saw a great variation with some groups reaching the relational level of SOLO by performing extensive research (e.g, looking up how much carbon dioxide does a person produce) and hypothesizing about possible outcomes of the model. For example, when describing what do they hope to observe form their model, the VWO group exploring the optimal strategy for potato farming (case Potato farm 2) wrote, "*we hope to find out what factors help increase the profit and what factors do not influence it that much. [...] Out hypothesis is that fertilizer will really increase the profit but*

	What do you know?	Modeling: what?	Observing: what?	Design and implementation	Validation	Experiment	Analysis	Answer	Reflection
Total	1.38	1.43	1.68	2.43	2.08	1.58	1.64	1.66	1.77
HAVO	0.88	0.88	1.25	2.25	1.50	0.57	0.71	0.57	1.63
VWO	1.52	1.59	1.79	2.48	2.24	1.83	1.86	1.93	1.81
t-test significance	**0.031**	**0.031**	**0.049**	**0.188**	**0.012**	**0.002**	**0.007**	**0.002**	**0.284**

Table 1: Mean scores and significance levels of differences in the performance of the HAVO groups compared to the VWO groups.

that pesticides have little influence on it. To prevent diseases and pests, we believe it is important to have large distance between the plants and to remove affected plants immediately." They continue in their logbook, "*We discussed what aspects of the phenomenon influence the profit. We decided not to take into account the seasons, weather and water. We did this because we look at the ideal conditions and other factors will not have a big influence.*"

Designing and implementing the model. For this part of the assignment we see the highest scores achieved and small variation among the classes. Every group managed to design a model and more than half of them arrived at a model matching or exceeding our minimal expert model. A small number of groups, while succeeding in writing some code, did not manage to write a meaningful program and subsequently failed to use it to perform an experiment. An example is the VWO group working on case Flood who stated their case and research question but failed to implement their model properly.

Validate the model. Here we see a great variation in scores and the VWO groups outperforming the HAVO groups significantly. From the HAVO groups only one reached relational level and one group did nothing meaningful to validate their model. The response from the Cheese barn 3 group is exemplary: "*The agents simulate the production and sales of a cheese barn, thus the cheese production, the sale of cheese, the agents simulated the production and sale of cheese as it happens in the real world, so this corresponds well with each other.*" All the VWO groups validated their models to some extent and almost a third of them reached the relational level, e.g. saying in case of life on Mars (case Mars 4), "*... maintenance of buildings and solar panels ... are not relevant for the results of our model but could play a small role for our research question.*"

Experiment, analysis and answering the research question. Again, here we see a great variation in the scores with the VWO groups significantly outperforming the HAVO groups. While 15 out of 17 VWO groups performed an experiment and a number of them extensively employed the Behavior Space (a feature of NetLogo allowing for systematic parameter sweeping and recording the results of each model run), out of the eight HAVO groups only three provided evidence of performing an experiment and none of them used Behavior Space. The quality of the subsequent analysis of the results and the answers to the research question seem to be directly related to the quality of the experiment. Analysis and answering the research question are the only aspects of the assignment where a total of four VWO groups reached extended abstract level. The VWO group exploring the case of bank counters (case Bank 3) notes, "*assigning the tasks to specific counters leads to specialization of employees ... this division of labor leads to faster and more efficient performance: by performing the same task all the time, the employees become specialized in particular tasks allowing them to carry out these tasks quicker. Conversely, non-specialized employees will carry out their tasks slower: because they get varying tasks all the time, they lack specific knowledge and need to look up things for the customers, causing the task to last longer. Because of this, tasks in scenarios 2 and 3 would take longer. However, the question is to what extent does this counterbalance the efficient engagement of the bank counters. This, then, is something that would require further research.*"

Reflection. The HAVO groups and VWO groups perform similarly. Interestingly, none of the groups in the 12^{th} grade VWO reached relational or extended abstract level. One of the 11^{th} grade VWO groups reaching that relational level explored evacuation of a burning building (case Fire 1) and wrote in their wish list, "*What we'd like to implement in our model is turtles making a clear choice what emergency exit to take. ... Sometimes they seem to doubt what exit to take, walk back and forth between the exits and eventually wait for to long — dead through fire. That's a pity because in reality, they could've survived.*"

Logbooks. In one of the schools the students were asked to keep a logbook, and in the other they were not, so the logbooks were not assessed separately. However, following the assessment finetuning guidelines, we read the submitted logbooks looking for evidence of answers to other questions.

6 DISCUSSION AND CONCLUSION

We designed and investigated an assessment instrument for the assessment of the intended learning outcomes for computational science. The design process was all but straightforward due to the fact that some ILO's of modeling are at the core of CS (e.g. implementation of the model), while others are not often seen in a CS classroom (e.g. experiment). Even for implementation, which comes down to programming, it was not easy to find related work

addressing assessment of programs at just the right level of abstraction. The same holds true for validation: while there is plentiful literature on validation of computational models, we could not find any focusing on the assessment of validation in a formal learning setting.

The project documentation and program code proved to be sufficient sources for assessment. However, when possible, we suggest to let students present their projects in the classroom too, and we encourage the teachers using this assessment instrument to take into account their observations of students at work when assessing their projects, as suggested by a number of teachers participating in one of our previous studies [11]. Indeed, the teacher who cooperated in this research noted that, while assessing his students' work, he constantly thought of his impressions from the classroom and wanted to take these impressions into account. This might be especially important for students who perform poorly when verbalizing their thoughts, as witnessed with many HAVO students in the parts of the assignment requiring textual descriptions such as stating the case and research question. We saw that none of these students achieved extended abstract level, while in the 12^{th} grade VWO one teacher found four instances of student groups reaching it. The other teacher, however — while assessing the same projects — found none and said, "*it was difficult to see clearly where the boundary lies between relational and extended abstract levels.*" Therefore, it could be argued that the extended abstract level is unobtainable for HAVO students, which would signify a situation similar to the one described by Castro and Fisler who found no instances of extended abstract level in their students' work [8]. Meerbaum-Salant et al. did not consider it at all and designed assessment with only the three intermediate SOLO categories to monitor novices' learning of CS concepts [24]. An issue to consider here is the question, what level of understanding is intended for the HAVO students, as opposed to the VWO students, and with what purpose are the students learning about computational science. The HAVO students are following education stressing a hands-on approach and leading to higher professional education and are often described as *thinking actors* [2], which could explain why there is no significant difference in their performance levels — compared to those of the VWO students — when designing and implementing models. The VWO students — often described as *acting thinkers* [2] — prepare for universities in an educational setting embracing a scientific frame of mind and it is not surprising that they significantly outperform HAVO students when validating their models and using them to conduct research — i.e. perform experiments, analyze results and draw conclusion. Therefore, we want to encourage teachers using this instrument to put more emphasis on the aspects of the modeling process related to the specific needs of their students. Arguably, for the HAVO students it might be more important to get a clear picture of the phenomenon being modelled and focus on the development — or possibly only enhancement — of a model, while for the VWO students, with the whole of their education emphasizing the scientific attitude, it might be more important to view a model as a vehicle to engage in scientific research and develop and use it as such. In order to cater to their needs, we repeat our recommendation to further sharpen the instruction about experimentation and data analysis [17] and add a suggestion to actively coach students in the first phases of their modeling projects when stating the case and research question and performing the accompanying research.

In conclusion, our assessment instrument in the form of a practical assignment and accompanying rubrics based on the SOLO taxonomy proved to be reliable, as indicated by a high rate of inter-rater agreement. Its validity is corroborated by exposing the significant differences in the performance levels of the HAVO students compared to the VWO students: as expected, the performance levels of the VWO students were significantly higher for almost all the criteria.

The results of this study exposed the needs of specific groups of students to receive instruction prior to and during their work on the assignments, and they informed us about the shortcomings in the curriculum intervention. All of these findings will contribute to the further refinement of the instrument itself, to the development of the teaching materials — an effort that will be reported elsewhere — and to the development of the CS curriculum in secondary education in the Netherlands, CS teacher training and CS education in general.

ACKNOWLEDGMENTS

This work is supported by the The Netherlands Organisation for Scientific Research grant nr. 023.002.138.

REFERENCES

[1] Edouard Amouroux, Benoit Gaudou, Stéphanie Desvaux, and Alexis Drogoul. 2010. Odd: A promising but incomplete formalism for individual-based model specification. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on.* IEEE, 1–4.

[2] E. Barendsen and J. Tolboom. 2016. *Advisory report (intended) curriculum for informatics for upper secondary education.* Technical Report. SLO. ID: 1204.

[3] Bernhard Bauer, Jrg P. Mller, and James Odell. 2001. Agent UML: A formalism for specifying multiagent software systems. *International journal of software engineering and knowledge engineering* 11, 03 (2001), 207–230.

[4] John Biggs and Catherine Tang. 2011. *Teaching for quality learning at university.* McGraw-Hill International. ID: 875.

[5] P. Blikstein and U. Wilensky. 2009. An Atom is Known by the Company it Keeps: Content, Representation and Pedagogy within the Epistemic Revolution of the Complexity Sciences. (2009). ID: 784.

[6] Karen Brennan and Mitchel Resnick. 2012. New Frameworks for Studying and Assessing the Development of Computational Thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada.* ID: 763.

[7] Michael E. Caspersen and Palle Nowack. [n. d.]. Model–Based Thinking & Practice. ([n. d.]). ID: 1014.

[8] Francisco Enrique Vicente Castro and Kathi Fisler. 2017. Designing a Multifaceted SOLO Taxonomy to Track Program Design Skills Through an Entire Course. In *Proceedings of the 17th Koli Calling Conference on Computing Education Research (Koli Calling '17).* ACM, New York, NY, USA, 10–19. https://doi.org/10.1145/3141880.3141891

[9] CSTA Computational Thinking Task Force. 2011. Operational Definition of Computational Thinking for K–12 Education. http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf ID: 744.

[10] J. K. Gilbert. 2006. On the nature of "context" in chemical education. *International Journal of Science Education* 28, 9 (2006), 957–976. ID: 343.

[11] N. Grgurina, E. Barendsen, C. Suhre, K. van Veen, and B. Zwaneveld. 2017. Investigating Informatics Teachers' Initial Pedagogical Content Knowledge on Modeling and Simulation. (2017).

[12] N. Grgurina, E. Barendsen, C. Suhre, K. van Veen, and B. Zwaneveld. 2018. Assessment of Modeling Projects in Informatics Class. in press.

[13] Nataša Grgurina, Erik Barendsen, Klaas van Veen, Cor Suhre, and Bert Zwaneveld. 2015. Exploring Students' Computational Thinking Skills in Modeling and Simulation Projects: a Pilot Study. In *Proceedings of the Workshop in Primary and Secondary Computing Education
.* ACM, 65–68. ID: 1126.

[14] Nataša Grgurina, Erik Barendsen, Bert Zwaneveld, Wim van de Grift, and Idzard Stoker. 2013. Computational Thinking Skills in Dutch Secondary Education. In

Proceedings of the 8th Workshop in Primary and Secondary Computing Education. ACM, 31–32. ID: 1072.

[15] Nataša Grgurina, Erik Barendsen, Bert Zwaneveld, Klaas van Veen, and Idzard Stoker. 2014. Computational Thinking Skills in Dutch Secondary Education: Exploring Pedagogical Content Knowledge. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*. ACM, 173–174. ID: 1069.

[16] Nataša Grgurina, Erik Barendsen, Bert Zwaneveld, Klaas van Veen, and Idzard Stoker. 2014. Computational Thinking Skills in Dutch Secondary Education: Exploring Teacher's Perspective. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*. ACM, 124–125. ID: 1056.

[17] Nataša Grgurina, Erik Barendsen, Bert Zwaneveld, Klaas van Veen, and Cor Suhre. 2016. Defining and Observing Modeling and Simulation in Informatics. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. Springer, 130–141. ID: 1378.

[18] Volker Grimm, Uta Berger, Finn Bastiansen, Sigrunn Eliassen, Vincent Ginot, Jarl Giske, John Goss-Custard, Tamara Grand, Simone K. Heinz, and Geir Huse. 2006. A standard protocol for describing individual-based and agent-based models. *Ecological Modelling* 198, 1-2 (2006), 115–126.

[19] Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, and Steven F. Railsback. 2010. The ODD protocol: a review and first update. *Ecological Modelling* 221, 23 (2010), 2760–2768. ID: 1182.

[20] Rosária S. Justi and John K. Gilbert. 2002. Science teachers' knowledge about and attitudes towards the use of models and modelling in learning science. *International Journal of Science Education* 24, 12 (2002), 1273–1292. ID: 936.

[21] David R. Krathwohl. 2002. A revision of Bloom's taxonomy: An overview. *Theory into practice* 41, 4 (2002), 212–218. ID: 789.

[22] Sze Yee Lye and Joyce Hwee Ling Koh. 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior* 41 (2014), 51–61.

[23] S. Magnusson, J. Krajcik, and H. Borko. 1999. *Nature, sources, and development of pedagogical content knowledge for science teaching*. Kluwer, 95–132. ID: 584.

[24] Orni Meerbaum-Salant, Michal Armoni, and Mordechai Ben-Ari. 2013. Learning Computer Science Concepts with Scratch. *Computer Science Education* 23, 3 (2013), 239–264. ID: 1080.

[25] Birgit Müller, Stefano Balbi, Carsten M Buchmann, Luís De Sousa, Gunnar Dressler, Jürgen Groeneveld, Christian J Klassert, Quang Bao Le, James DA Millington, Henning Nolzen, et al. 2014. Standardised and transparent model descriptions for agent-based models: Current status and prospects. *Environmental Modelling & Software* 55 (2014), 156–163.

[26] James J. Odell, H. Van Dyke Parunak, and Bernhard Bauer. 2000. Representing agent interaction protocols in UML. In *International Workshop on Agent-Oriented Software Engineering*. Springer, 121–140.

[27] K. Van Overveld, T. Borghuis, and E. van Berkum. 2015. *From Problems to Numbers and Back*. Eindhoven University of Technology, Eindhoven. ID: 1099.

[28] James Rumbaugh, Ivar Jacobson, and Grady Booch. 2004. *Unified modeling language reference manual, the*. Pearson Higher Education.

[29] Frits Vaandrager. 2011. A First Introduction to uppaal. *Deliverable no.: D5.12 Title of Deliverable: Industrial Handbook* 18 (2011).

[30] Jacqueline Whalley, Tony Clear, Phil Robbins, and Errol Thompson. 2011. Salient elements in novice solutions to code writing problems. In *Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114*. Australian Computer Society, Inc., 37–46.

[31] U. Wilensky. 1997. NetLogo Wolf Sheep Predation model. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation

[32] U. Wilensky. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/

[33] Uri Wilensky. 2014. Computational thinking through modeling and simulation. *Whitepaper presented at the summit on future directions in computer education.Orlando, FL.http://www.stanford.edu/ coopers/2013Summit/WilenskyUriNorthwesternREV.pdf* (2014). ID: 1120.

[34] Uri Wilensky and William Rand. 2015. *An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo*. MIT Press. ID: 1184.

[35] Kenneth Wolf and Ellen Stevens. 2007. The role of rubrics in advancing and assessing student learning. *The Journal of Effective Teaching* 7, 1 (2007), 3–14. ID: 975.

[36] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. 2016. An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research* 53, 4 (2016), 562–590.