

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/199856>

Please be advised that this information was generated on 2019-06-02 and may be subject to change.

Assessment of Modeling Projects in Informatics Class

Natasa Grgurina, *n.grgurina@rug.nl*

University of Groningen, Netherlands

Erik Barendsen, *e.barendsen@cs.ru.nl*

Radboud and Open University, Netherlands

Cor Suhre, *c.j.m.suhre@rug.nl*

University of Groningen, Netherlands

Klaas van Veen, *klaas.van.veen@rug.nl*

University of Groningen, Netherlands

Bert Zwaneveld, *zwane013@planet.nl*

Open University, Netherlands

Abstract

The introduction of the new Informatics curriculum in the Netherlands in 2019 raises the need for new teaching material that includes 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 and an assessment instrument consisting of a practical assignment and grading rubrics to assess student's level of understanding. The rubrics we developed can be used both for formative and summative assessment. In this paper we describe the design of this assessment instrument and indicate further research directions focusing on validation of this instrument.

Keywords

modeling and simulation; NetLogo; assessment; SOLO-taxonomy

Introduction

In the Netherlands, where informatics 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 informatics 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 a different 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.*" (Barendsen & Tolboom, 2016). 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 and we both participate in these endeavors and monitor the developments.

This study is a part of a larger research project on teaching Computational Science in the context of informatics in Dutch secondary education, investigating pedagogical aspects and 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. (Magnusson, Krajcik, & Borko, 1999), 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) (Grgurina, Barendsen, Zwaneveld, van de Griff, & Stoker, 2013), made initial explorations of teachers' PCK (Grgurina, Barendsen, Zwaneveld, van Veen, & Stoker, 2014a; Grgurina, Barendsen, Zwaneveld, van Veen, & Stoker, 2014b) and of the computational modeling process (Grgurina, Barendsen, van Veen, Suhre, & Zwaneveld, 2015), 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, and established what data sources were suitable for assessment — Magnusson's element M4 (Grgurina, Barendsen, Zwaneveld, van Veen, & Suhre, 2016), and finally, investigated teachers' initial pedagogical content knowledge on modeling and simulation (Grgurina, Barendsen, Suhre, van Veen, & Zwaneveld, 2017). In our subsequent study, we focus on monitoring the levels of understanding in the learning outcomes of students engaging in modeling projects - Magnusson's element M4 - and address the following research question: *What are the characteristics of the assessment instrument for assessment of the intended learning outcome for computational science?* In this paper, we describe the design of this assessment instrument. The results of the entire study will be reported elsewhere.

Background and Related Work

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) (CSTA Computational Thinking Task Force, 2011). With the arrival of computers into schools, new venues are created to aid students' learning in various disciplines through the use of computer models (Blikstein & Wilensky, 2009; Van Overveld, Borghuis, & van Berkum, 2015). 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" (2014), as do Caspersen and Nowack (2013). Indeed, modeling plays a significant role in the development and learning of science (Justi & Gilbert, 2002) and informatics 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 (Gilbert, 2006) and informatics. In the Informatics 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 (Grgurina et al., 2016).

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 as they program, namely "*sequences, loops, parallelism, events, conditionals, operators, and data*"; computational practices describing the practices designers develop as they program, 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*". (Brennan & Resnick, 2012). 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*" (Zhong, Wang, Chen, & Li, 2016). Using this framework, Lye and Koh analyzed 27 intervention studies in K-12 aimed at the development of computational thinking 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.” (Lye & Koh, 2014) Since assessments can provide learning opportunities, 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. (Brennan & Resnick, 2012).

These views are corroborated by the findings in our prior study on informatics teachers’ pedagogical content knowledge (PCK) of modeling and simulation, where we learned that the interviewed teachers mostly suggest 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, yet very few corresponding quality indicators used to judge to what extent these criteria are met (Grgurina et al., 2017).

In the eyes of the students, the assessment defines the actual curriculum, according to Biggs and Tang, who advocate a criterion-referenced system where the objectives are imbedded in the assessment tasks. 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. The learning outcomes can be classified using the Structure of the Observed Learning Outcome (SOLO) which describe 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: (Biggs & Tang, 2011). Meerbaum-Salant et al. interpreted 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.

According to them, 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*”, so they combined the Bloom’s taxonomy and the three intermediate categories of the SOLO taxonomy in order to assess how novice programmers learned programming with Scratch (Meerbaum-Salant, Armoni, & Ben-Ari, 2013). 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*”. To alleviate this problem, they developed a mapping framework where first, the salient elements are identified at syntactic level of the code; subsequently, basic replicable and discernible features such as redundancy, efficiency, generalizability and integration are abstracted from the code itself, and finally, SOLO mapping takes place to the five SOLO categories they suggest for code writing solutions (Whalley, Clear, Robbins, & Thompson, 2011).

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 (Castro & Fisler, 2017). 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 (Biggs & Tang, 2011). Assignments for complex tasks encompassing diverse ILOs — such as 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.

Assessment instrument

Based on these findings, we developed constructionist teaching material about agent-based modeling with NetLogo, meant for the informatics students in the 11th and 12th grades 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 (Grgurina et al., 2016), and focuses not only on computational concepts such as programming to implement the model, but also on 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 suggestions for the rubrics construction by Wolf and Stevens (2007), from the modeling cycle we first identified the criteria that defined performance as: stating the case and the research question, designing the model, implementation, validation, experiment, analysis, answering the research question, reflection, and additionally, logbooks. Subsequently, we designed an assessment instrument consisting of 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 rubric based on SOLO taxonomy with unequally weighted criteria defining performance. The description of SOLO categories was based on the interpretation by Meerbaum-Salant et al., stressing the progression from the local to the global perspective.

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.

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. 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 Wilensky & Rand (2015).)

Design the model. Design a model following the questions listed here. Describe 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 Wilensky & Rand (2015).)

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 your 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.

In addition, the students were asked to keep a logbook recording all their activities, problems, successes and dead ends they encountered; possible explanations for problems and successes, and finally, lessons learned.

Grading Rubrics

After we identified the criteria that defined performance, we created performance descriptions (Wolf & Stevens, 2007) to describe the appropriate level of understanding for intended learning outcomes (Biggs & Tang, 2011). Here we quote some of these descriptions:

Case and research question

- **Prestructural:** (1) Nothing or simplistic idea of the phenomenon. Performed no research. (2) Nothing, or a few non-specific remarks but missing the point (3) Research question not clear
- **Unistructural:** (1) Some general description. Performed no research or only limited to isolated aspects of the phenomenon (2) Few isolated aspects of the phenomenon identified. (3) Identified the question from a local perspective.
- **Multistructural:** (1) Performed some research. Able to name more relevant aspects of the phenomenon, but mentions no relations among these aspects (2) Described what (part of the) phenomenon is being modeled. (3) Described the question from a multi-point perspective.
- **Relational:** (1) Performed research. Complete idea of the phenomenon. Able to name relevant aspects of the phenomenon, have insight into relations among these aspects (2) Described what (part of the) phenomenon is being modeled. (3) The research question clear and predicts possible outcomes.
- **Extended abstract:** (1) 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) Additionally, theorize about possible generalization of the model or transfer into a different context. (3) 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.

Results and Further Research Direction

In this paper, we described the design of our assessment instrument for the assessment of the intended learning outcomes for Computational Science consisting of a practical assignment covering the ILO's defining Computational Science and an accompanying rubric based on SOLO taxonomy that describes the levels of understanding. During the design process, we faced many challenges due to the fact that some ILO's of modeling are at the core of informatics (e.g. implementation of the model), while others are not often seen in an informatics classroom (e.g. experiment). Even for implementation, which comes down to programming, it was not easy to find related work addressing assessment of programming at just the right level of granularity. The same holds true for validation: while there is plentiful literature on validation of computational models, to our best knowledge there is none focusing on the assessment of validation in a formal learning setting.

So far, several teachers used our teaching material and assessment instrument to teach Computational Science in the Informatics class of the 11th and 12th grade of pre-university education (VWO) in the Netherlands. We are collecting and analyzing feedback from them and their students in order to aid the on-going project of development of teaching materials and assessment instruments. Specifically, the current version of the assessment instrument will be analyzed to establish its reliability, validity and objectivity, and in particular, it will be scrutinized in relation to the descriptions and attainability of currently proposed levels of understanding as specified in the rubrics.

Acknowledgments

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

References

- Barendsen, E., & Tolboom, J. (2016). Advisory report (intended) curriculum for informatics for upper secondary education. Enschede: SLO.
- Biggs, J., & Tang, C. (2011). Teaching for quality learning at university. McGraw-Hill International.
- Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: Content, representation and pedagogy within the epistemic revolution of the complexity sciences.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada,
- Caspersen, M. E., & Nowack, P. (2013). Model-Based thinking & practice.
- Castro, F. E. V., & Fisler, K. (2017). Designing a multi-faceted SOLO taxonomy to track program design skills through an entire course. Proceedings of the 17th Koli Calling Conference on Computing Education Research, pp. 10-19.
- CSTA Computational Thinking Task Force. (2011). Operational definition of computational Thinking for K-12 education. Retrieved 10/16, 2013, from <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>

- Gilbert, J. K. (2006). On the nature of "context" in chemical education. *International Journal of Science Education*, 28(9), 957-976.
- Grgurina, N., Barendsen, E., Suhre, C., van Veen, K., & Zwaneveld, B. (2017). Investigating informatics teachers' initial pedagogical content knowledge on modeling and simulation. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pp. 65-76.
- Grgurina, N., Barendsen, E., van Veen, K., Suhre, C., & Zwaneveld, B. (2015). Exploring students' computational thinking skills in modeling and simulation projects: A pilot study. *Proceedings of the Workshop in Primary and Secondary Computing Education*, pp. 65-68.
- Grgurina, N., Barendsen, E., Zwaneveld, B., van de Grift, W., & Stoker, I. (2013). Computational thinking skills in Dutch secondary education. *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, pp. 31-32.
- Grgurina, N., Barendsen, E., Zwaneveld, B., van Veen, K., & Stoker, I. (2014a). Computational thinking skills in Dutch secondary education: Exploring pedagogical content knowledge. *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*, pp. 173-174.
- Grgurina, N., Barendsen, E., Zwaneveld, B., van Veen, K., & Stoker, I. (2014b). Computational thinking skills in Dutch secondary education: Exploring teacher's perspective. *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, pp. 124-125.
- Grgurina, N., Barendsen, E., Zwaneveld, B., van Veen, K., & Suhre, C. (2016). Defining and observing modeling and simulation in informatics. *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, pp. 130-141.
- Justi, R. S., & Gilbert, J. K. (2002). Science teachers' knowledge about and attitudes towards the use of models and modelling in learning science. *International Journal of Science Education*, 24(12), 1273-1292.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61.
- Magnusson, S., Krajcik, J., & Borko, H. (1999). Nature, sources, and development of pedagogical content knowledge for science teaching. In J. Gess-Newsome, & N. G. Lederman (Eds.), *Examining pedagogical content knowledge* (pp. 95-132) Kluwer.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264.
- Van Overveld, K., Borghuis, T., & van Berkum, E. (2015). From problems to numbers and back. Lecture notes to 'A discipline-neutral introduction to mathematical modelling'. Eindhoven: Eindhoven University of Technology.
- Whalley, J., Clear, T., Robbins, P., & Thompson, E. (2011). Salient elements in novice solutions to code writing problems. *Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114*, pp. 37-46.
- Wilensky, U. (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](http://www.Stanford.Edu/~coopers/2013Summit/WilenskyUriNorthwesternREV.Pdf).
- Wilensky, U., & Rand, W. (2015). *An introduction to agent-based modeling: Modeling natural, social, and engineered complex systems with NetLogo* MIT Press.
- Wolf, K., & Stevens, E. (2007). The role of rubrics in advancing and assessing student learning. *The Journal of Effective Teaching*, 7(1), 3-14.
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53(4), 562-590.