Motivating Teachers to Teach Computing in Middle School A Case Study   of a Physical Computing Taster Workshop for Teachers

Aldo von Wangenheim[1]

Christiane Gresse von Wangenheim[1]

Fernando S. Pacheco[2]

Jean C. R. Hauck[1]

Miriam Nathalie F. Ferreira[1]

[1]Federal University of Santa Catarina

[2]Federal Institute of Santa Catarina

Abstract

Computing education in schools faces several problems, such as a lack of computing teachers and time in an already overloaded curriculum. A solution can be a multidisciplinary approach, integrating the teaching of computing within other subjects, creating the need to motivate teachers from other disciplines to teach computing in middle school. Therefore, the motivation and training of in-service teachers becomes crucial, as they need to have computing content and technological knowledge as well as pedagogical content knowledge. Yet, so far there exist very few training programs. Thus, as part of a comprehensive outreach program, we present a study on a one-day taster workshop for middle school teachers on physical computing education. Participants learn computer programming practice and computational thinking by programming an interactive robot. The workshop also approaches pedagogical aspects for teaching computing and technical issues regarding the installation and preparation of the required hardware/software. Preliminary results of its application with public school teachers in Florianopolis/Brazil are positive, motivating the majority of participants to introduce computing into their classes. However, our results also highlight that in order to enable teacher to apply the workshops effectively, longer training courses and ongoing support is required.

Keywords: teacher training, physical computing, middle school, programming

1. Introduction

1.1 Why is computing important?

Computer science is driving job growth and innovation. Rapid technology advancements and the growing number of professions that rely on computer science make it crucial for all students to have opportunities to become not only IT literate (capable to use today's technology), but to acquire IT fluency, adding the capability to independently learn and use new technology as it evolves, including the active use of computing (CSTA, 11). Learning computing encourages students to create and innovate, and allows them to take advantage of future career opportunities. Therefore, it is important to provide opportunities for children to learn computing in school (Naughton, 12).

1.2 Teaching computing in schools

Pre-college exposure to computing is a way to generate interest and curiosity, as well as to establish a sense of competency that may lead to further interest in this area (Google & Gallop, 15). Currently, schools are starting to incorporate computing education in many ways: dedicated or multidisciplinary school classes, after-school groups or clubs or as online courses, in the most diverse ways, e.g., by programming games, animations, or through physical computing/robotics integrating the virtual into the real world (Lye & Koh, 14). And, although,

more resources are becoming available (including age appropriate programming environments (such as Scratch (MIT, 17) or Snap! (Harvey & Mönig, 17)), curriculum guidelines, lesson plans and teaching materials), computing education is still not widely accessible to students throughout their K-12 education including middle school (Google & Gallup, 15).

1.3 Need for teachers qualified for computing education

Implementing computing education at school demands motivated and dedicated teachers with computing, pedagogical and technical competences in order to teach computing in a way that truly engages students (Gal-Ezer & Stephenson, 10)(Goode, 08). However, the problem is that computing school teachers are very scarce (Google & Gallop, 15), as the incentive to pursue lucrative careers in IT industry is much higher than the incentive to teach (Century et al., 13). Thus, computing education may have to be provided by in-service teachers holding credentials in other disciplines (Qualls & Sherrell, 10).

1.4 Training teachers to teach computing

In this context, in-service teacher training in computing is crucial to provide students' access to quality education (Giannakos et al., 14) (Gal-Ezer & Stephenson, 10). Teachers are required to have a broad knowledge not only of subject matter content and technological knowledge (CSTA, 17), but also pedagogical content knowledge to be able to integrate the teaching of computing competencies effectively into their teaching practice (Mishra & Koehler, 06)(Shulman, 86). Thus, in order to amplify computing education in a multidisciplinary approach, it is important to support teachers with rich professional development programs (Goode, 08). A solution is for universities to provide computing related training to school teachers so that these teachers can pass on their knowledge and inspiration to their students (Liu et al., 11a).

1.5 Related work

In recent years, several universities have experimented with the idea of school teacher outreach programs by means of training workshops, professional networks, and continued education (Liu et al., 11a). Training programs for in-service teachers typically aim to disseminate curriculum modules that school teachers can implement in the classroom (Blum & Cortina, 07). Typically computing concepts are taught, mostly programming, also covering, to a lesser extend, curriculum aspects (Liu et al., 15)(Liu et al., 11a). Aiming at computing education on K-12 level including middle school, many training programs focus on visual block-based programming languages such as Scratch, Alice (Liu et al., 11b) or Snap! (Liu et al., 15) to develop games or animations. Few programs cover other languages such as Python (Partanen et al., 16) or involve robotics, typically using LEGO Mindstorms (Jackson et al., 14) (Kim et al., 12) (Alimisis et al., 09), inhibiting large-scale applications in Brazilian schools due to their costs (Ruzzenente et al., 12). The majority of the courses are offered in a traditional way (face-to-face) in week long workshops (Liu et al., 15) (Jackson et al., 14), or as online courses (Partanen et al., 16). Yet, most of the existing training programs focus exclusively on high school teachers (Cooper et al., 10)(Lauwers et al., 10)(Bruckman et al., 09)(Blum & Cortina, 07)(Ni et al., 11). Fewer programs approach middle school teachers (Jackson et al., 14). Another shortcoming is often their exclusive focus on computing knowledge, not teaching pedagogical aspects in order to indicate tools to apply computing in their classes. Furthermore, in our particular context in Brazil, some results are not readily applicable due to differences in the middle school curriculum as well as the need for customized teaching units in Brazilian Portuguese. Given this situation, the purpose of this article is to explore a new taster workshop to motivate and enhance middle school teachers' capabilities to teach computing in a multidisciplinary way.
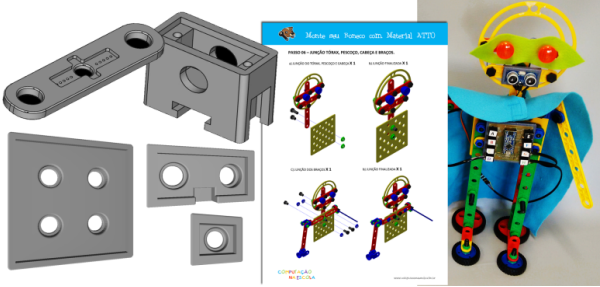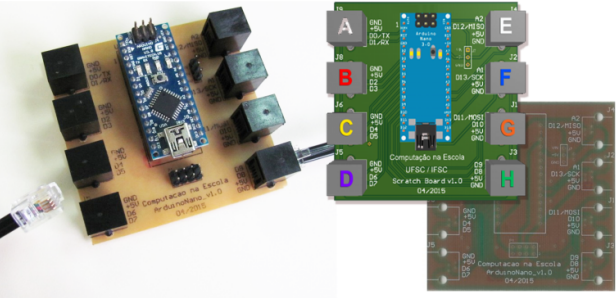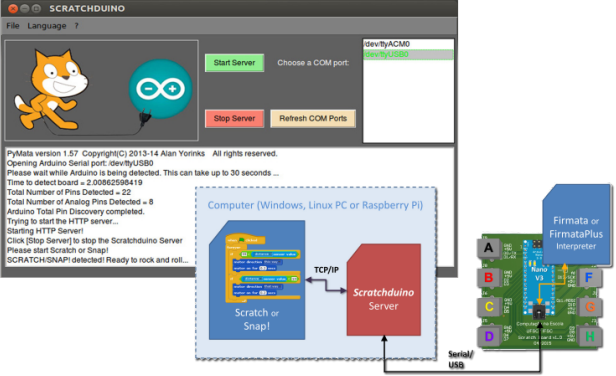
2. Programming a Superhero Robot: Physical Computing Workshops in Middle Schools

Physical computing or robotics has gained increased attention as an inspiring way to teach computing by integrating the virtual into the real world (O'Sullivan & Igoe, 04). As part of the initiative *Computação na Escola* (http://www.computacaonaescola.ufsc.br), we designed a 3 1/2-hours workshop, in which middle school students learn how to program an interactive superhero robot (Gresse von Wangenheim et al., 2017). This involves learning the use of simple commands as well as events, conditionals, and loops, operators and interface commands to read sensors and pass over outputs as well as how to create and use a function. Basic computing
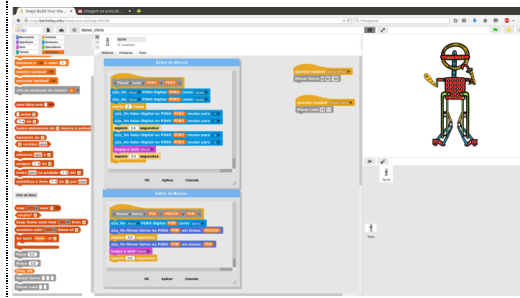
concepts such as understanding of algorithmic problem-solving (problem statement, implementation and testing cycle), collaboration in form of pair programming as well as the understanding what a computer program is are taught implicitly. Within the context of a narrative, workshop participants "give life" to a superhero robot, using sensors and LEDs (Figure 2). The workshop is hands-on, minimizing theoretical explanations with the students step-by-step programming their robot applying the presented concepts.

For the workshops we developed a low-cost, platform independent, reliable strategy, including an Arduino Nano microcontroller, hardware parts and the visual programming language Scratch or Snap! (Table 1).

Table 1. Elements used in the physical computing workshop

Physical structure of the robot using *ATTO Educacional* (http://www.attoeducacional.com.br), a set of multifunctional structural parts (including bolts, rivets, bars, etc.) injected into non-toxic, washable, colored plastic and set of ATTO-compatible 3D printable parts for attaching the automation components as well as assembling instructions.	
Electronic kit. To read the sensors, drive the motors and communicate with the programming platform, we developed a strategy that integrates a microcontroller (Arduino Nano) and connectors that are easy to use. Additional hardware parts include a 9g servomotor, an ultrasonic distance sensor and two LEDs. In order to facilitate the handling by the children, we developed the Scratchboard (http://www.computacaonaescola.ufsc.br/scratchboard), an open hardware break-out board that allows building physical devices using telephone cable jacks and helps visualizing the structure and logic of the connections.	
Scratchduino. In order to facilitate the establishment of the communication between Arduino and Scratch/Snap!, we developed a communication server (http://www.computacaonaescola.ufsc.br/scratchduino). It is an easy to use, multiplatform GUI-based program that acts as a bidirectional protocol translator, providing interoperability between the Scratch Extension Protocol and the traditional Firmata microcontroller protocol originally developed by Yorinks (2013).	

Portuguese localization for the Scratch and Snap! Arduino Blocks.	
Linux installation image enabling the easy installation and configuration of all required software.	
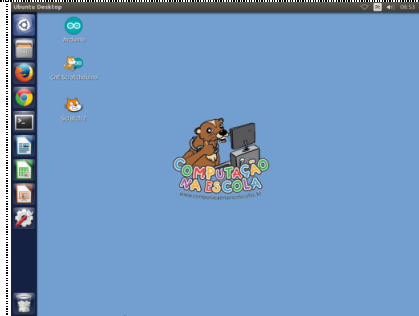
Diverse teaching materials are part of the unit (Table 2).
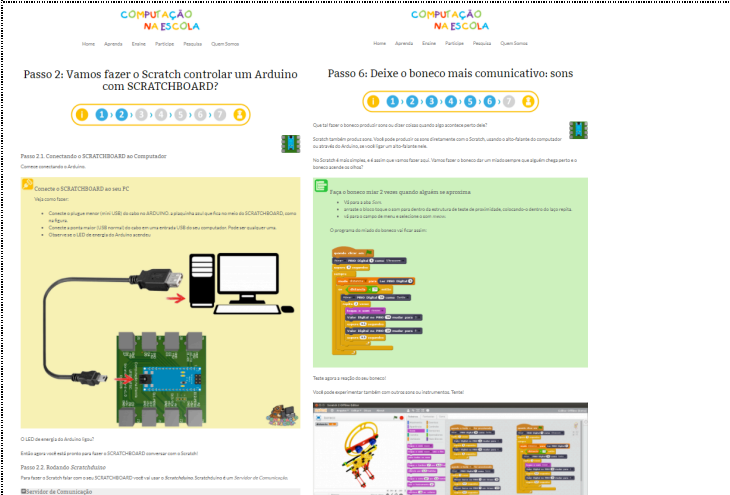
Table 2. Examples of teaching material

Syllabus defining the workshop learning objectives, sequence as well as the instructional strategies and materials.	

Slides in order to facilitate the teaching during the workshop presenting basic computing and physical computing concepts.	
Online classroom tutorial explaining step-by-step the programming of the robot as well as basic concepts. The tutorial can be used by the workshop participants in parallel to the instructor's explanations. It also offers a way to learn how to program the robot outside of the workshop.	
Online teacher tutorial and reference material explaining step-by-step how to prepare the school's computing lab, to install the Linux Installation Image, to prepare and upload the firmware for the microcontrollers and provides troubleshooting solutions for common problems.	
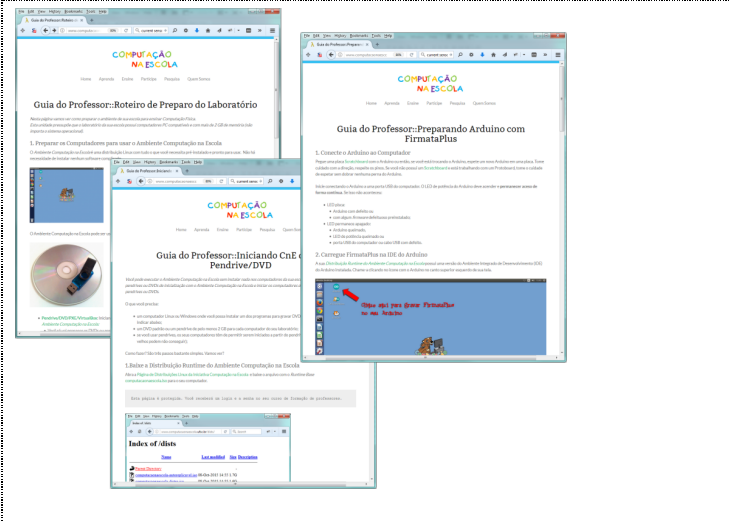
All materials are available under the Creative Commons license in Brazilian Portuguese at: http://www.computacaonaescola.ufsc.br/?page_id=1600&lang=en.

Several workshops were held either as a school unit or as family workshops with great success involving more than 150 participants (Gresse von Wangenheim et al., 2017b). The observed results indicate that workshop allows to learn basic computing concepts in an efficient, effective and fun way increasing interest in computing.
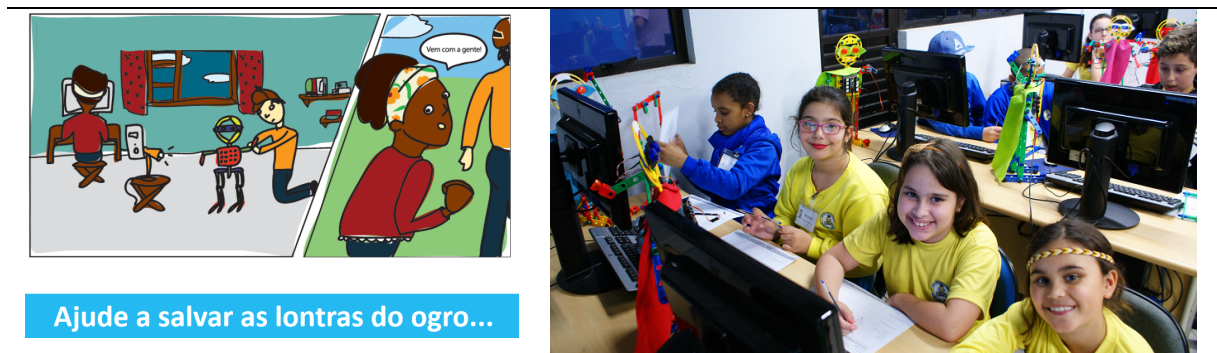
Figure 1. (a) Scene of the narrative; (b) students interacting with their programmed robots

3. Research Method

The objective of this research is the development, application and evaluation of an instructional unit for training in-service teachers for computing education in a multidisciplinary way in middle school. To achieve this goal, an exploratory case study was conducted to understand the phenomena observed during the application of the IU in a particular context and identify directions for future work (Fig. 2).
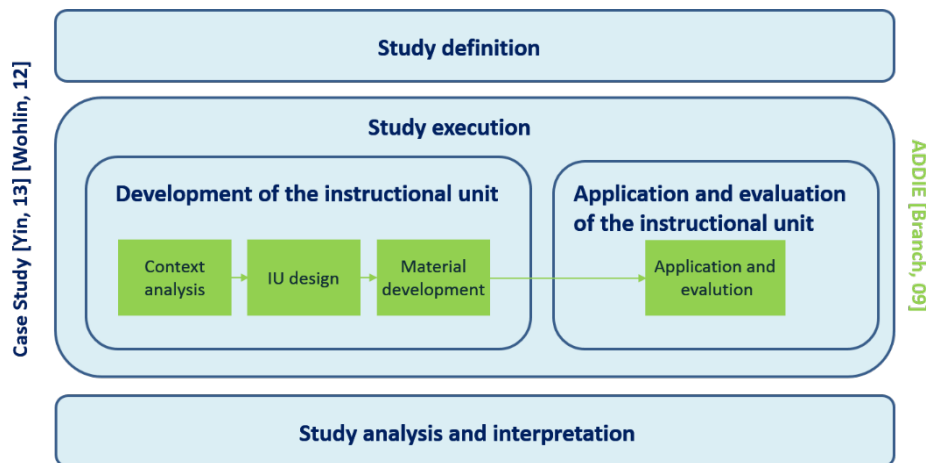
Figure 2. Research method

The case study is performed according to the procedure proposed by Yin (13) and Wohlin et al. (12): The study is defined in terms of objective, research questions and research design. From the objective, analysis questions and measures are systematically derived using the Goal/Question/Metric (GQM) approach (Basili et al., 94). Data collection instruments are defined with respect to the measures.

The execution of the study is carried out adopting ADDIE (Branch, 09) as instructional design approach. First, the instructional unit is developed. Therefore, the learners and the instructional environment are characterized. Learning needs are elicited and the learning objectives are defined. In accordance to the context, the instructional strategy is designed, defining its content, sequence and instructional methods to be adopted. Instructional material is developed in accordance to the instructional strategy. Then, the instructional unit is applied in training sessions and evaluated, collecting data as defined by the study definition.

The collected data is analyzed in relation to the research questions, using quantitative and qualitative methods. Then, the results are interpreted and discussed.

4. Physical Computing Taster Workshop for Middle School Teachers

4.1 Definition of the teacher workshop

As part of a comprehensive outreach program for training middle school in-service teachers, we designed a taster workshop in order to motivate and arise teachers' excitement for this area. The purpose of this workshop is to teach basic computing concepts to middle school teachers of any knowledge area (and/or teachers providing IT support in schools), visioning a multidisciplinary integration of computing into the existing curriculum.

It is expected that the teachers at least know how to use computers, but not necessarily have computing competencies. The workshop covers knowledge focusing on computational thinking and programming (loops, conditionals, event handling, etc.), the application of the software engineering cycle and collaborative practices as well as the understanding of what algorithms are and how they work. Using a visual blocks-based programming environment, learners should be able to describe what can be done with the environment as well as being able to use the environment to program an interactive robot. The teacher's workshop also covers the understanding of pedagogical content knowledge, including possibilities to integrate computing education in a multidisciplinary way in different disciplines. Furthermore, technical topics such as the installation and the preparation of the required hardware/software environment are approached, more specifically directed to teachers who provide IT support at the schools.

4.2 Design of the teacher workshop

The workshop is carefully designed to motivate and inspire middle school teachers, instilling the interest to integrate teaching computing competencies as part of their disciplines as well as to show them through hands-on experience of how physical computing can be taught in a multidisciplinary way. The one-day workshop contains separate sections following the teacher development model proposed by Shulman (00):

Motivation/contextualization: showing the importance and relevance of computing competencies for citizens in the 21th century and providing an overview on existing trends for teaching computing in middle schools.

Section 1. Computing Knowledge: focusing on computing competencies by teaching the physical computing workshop – Program your superhero robot. This includes the learning of basic physical computing concepts as well as the programming environment (Scratch or Snap!). We teach the programming of the robot step-by-step with a strong emphasis on learning by doing, always adopting a development cycle: identify requirements, mount hardware, program and test. In the end of this section, participants are invited to freely explore adaptations of the robot. By carrying out the workshop the same way as to be applied in the classroom, it also illustrates how the workshop can be taught.

Section 2. Pedagogical content knowledge: the instructional design of the workshop is presented as well as the instructional material available for the teachers. We also discuss typical teacher concerns related to adopting innovative content such as preparation time, students' motivation, and potential issues. Teachers also discuss how to integrate such a workshop into their specific disciplines based on given examples (Fig. 3).

Figure 3. Examples of multidisciplinary customizations of the interactive robot

Section 3. Technological knowledge. In order to be able to run the "Programming a Superhero Robot" workshop in the schools, it is necessary to have a hardware and software platform containing all tools and connections correctly configured and ready to use. Assuming that the teachers do not necessarily have a deep knowledge of operating systems and how to install the necessary tools to perform physical computing, we teach hands-on the installation and use of the Linux installation image.

Table 3. Syllabus of the teacher workshop

Content	Duration	Instructional method	Resources
Motivation/ contextualization	30 min	-Presentation - Discussion	- Slides -Video
Section 1. Computing Knowledge	210 min	- Practical activity following the step-by-step presentation of the instructor	- Scratch/ Snap! environment - Scratchduino - Robot (physical structure and hardware parts) - Slides - Online tutorial - Computers/projector
Measurement 1			- Participant post-unit questionnaire 1
Section 2. Pedagogical content knowledge	60 min	- Presentation - Discussion	- Example games - Scratch environment - Instructional material from history classes - Computers - Classroom projector
Section 3. Technological knowledge	90 min	- Practical activity in pairs with assistance of the instructors	- Scratch/ Snap! environment - Scratchduino - Robot (physical structure and hardware parts) - Slides - Online tutorial - Computers/projector
Debriefing	15 min	- Discussion	
Measurement 2			- Participant post-unit questionnaire 2

In accordance to the instructional strategy, the instructional materials have been developed and are available online in Brazilian Portuguese under the Creative Commons license (http://www.computacaonaescola.ufsc.br/?page_id=1960).

5. Application of the Physical Computing Taster Workshop for Middle School Teachers

The taster workshop was applied in an one-day training workshop with 19 middle school teachers from 16 public schools in Florianópolis/Brazil in 2016. The majority were women of 30-49 years. Most were IT support teachers who assist subject matter teachers with classes involving IT in diverse disciplines and maintain the IT environment at the schools. Most teachers did not have any experiences in teaching computing (Fig. 4).
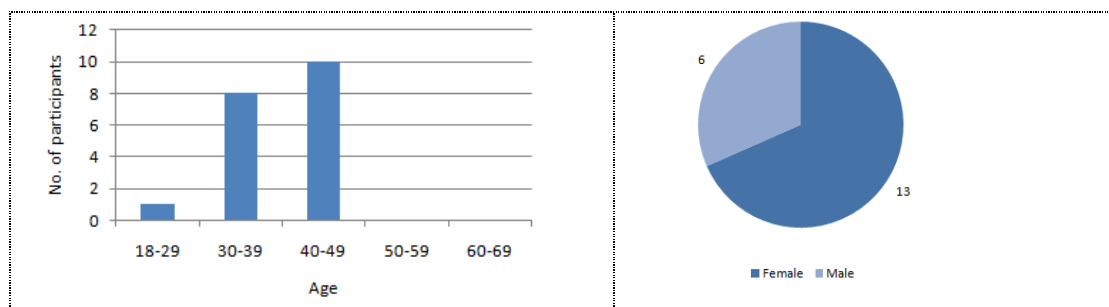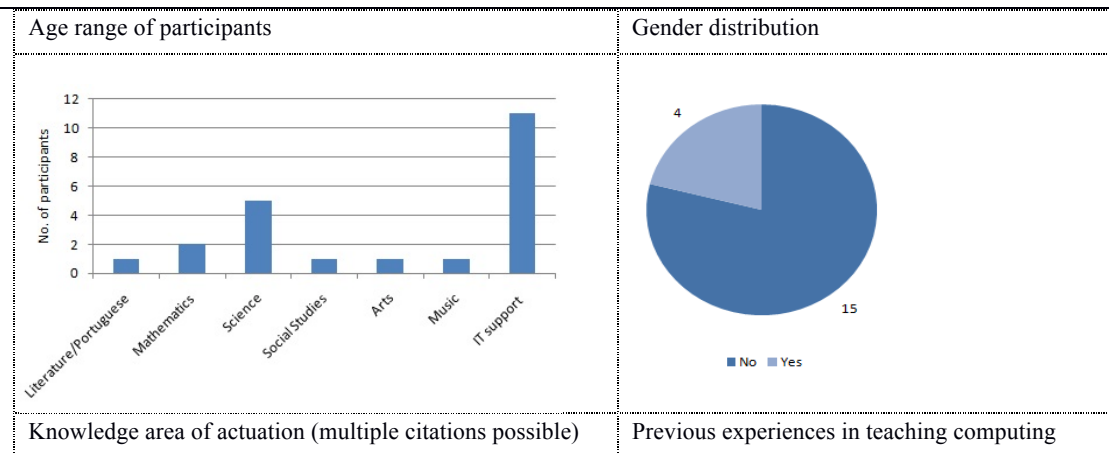
Figure 4. Demographic data on participating teachers

The classes were taught by university professors of the initiative *Computação na Escola* with computing and electronics background from the Federal University of Santa Catarina and the Federal Institute of Santa Catarina, supported by undergraduate students. During the workshop, the participants worked in pairs on the programming project.

Figure 5. Scenes from the workshop

6. Evaluation of the Physical Computing Taster Workshop for Middle School Teachers

6.1 Definition of the evaluation

The goal of this study is to explore and to understand aspects related to the workshop for motivating and teaching middle school teachers on physical computing. Based on this goal, the following analysis questions have been defined:

AQ1. Are the learning objectives achieved?

AQ2. Does the workshop facilitate learning?

AQ3. Does the workshop provide a positive perception of computing and motivate teachers to teach computing?

6.2 Data Analysis

Data has been collected from the participants after the first section and in the end (with respect to section 2 and 3) via questionnaires. The data was analyzed in a qualitative and quantitative way using descriptive statistics with respect to the analysis questions.

6.2.1 Are the learning objectives achieved?

All participants were able to use the visual programming environment Scratch with ease. As part of the robot-project, participants used the presented concepts and commands. All participants created the robot-themed physical computing project. Indirectly by programming the robot, the participants also learned how to use basic steps in algorithmic problem-solving. Starting from the specification of the robot, they implicitly followed a system engineering cycle by designing, programming and testing each part step-by-step. At the end of section 1 most participants believe that they can make computer programs (Fig. 6).
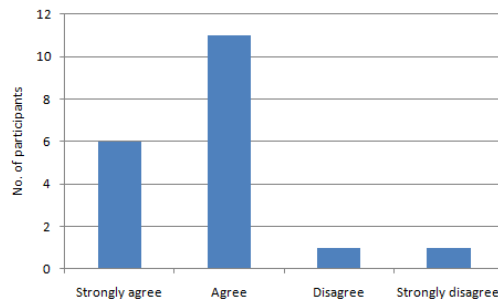
Figure 6. "I can make computer programs."

The participants also considered learning pedagogical content knowledge and technological knowledge useful (Fig. 7).
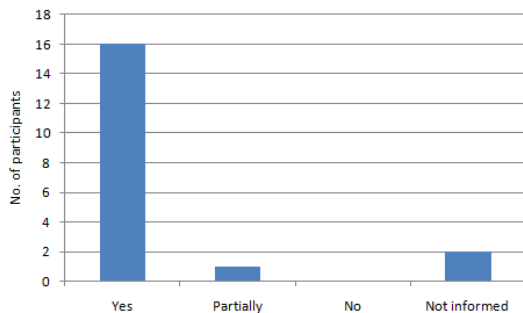
Figure 7a."Was learning pedagogical content knowledge useful?"
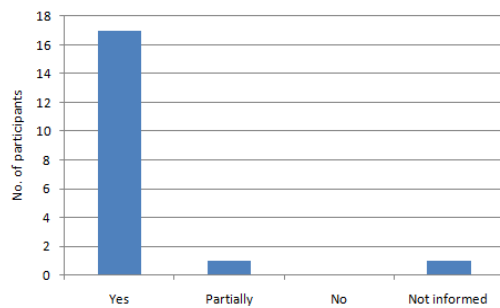
Figure 7b. "Was learning technological knowledge useful?"

The participants recognized that section 2 helped them to visualize possibilities on how to apply the workshop as part of their classes. They also emphasized the importance of section 3 (about technological aspects) (Table 4).

Table 4. Discursive comments on strengths of section 2 and 3 of the workshop

The electronics part, in which we use the Arduino.
Discussing the application in different knowledge areas.

Practical examples of using scratch and Arduino in schools.

Further information on programming.

Understanding of the required infrastructure and the software provided.

To think about application strategies at school, with the resources we have.

The question of how we can insert programming at school.

Learn how to prepare the infrastructure. Possibilities to make application at school feasible.

Know the Arduino board and realize that it is possible to configure it.

Knowing that information will be available on the site for consultation and study.

Difficult for me to evaluate. It was a lot of information. Perhaps the most difficult thing is going to have time to test, and to see if I can prepare everything.

Realize/reinforce the idea that it is necessary to start the work with the students with the acquired knowledge and then with time expand the learning.

Clarification about the position of devices and experiences already developed in Florianópolis.

Learning about Arduino

Everything was useful.

To show me how to set up Arduino.

Prepare the environment.

Understand that it is possible to search the available updates on the tutorials to develop the project.

Yet, although 5 teachers indicated that they felt sufficiently prepared to run such a workshop as part of their classes (Fig. 8), most indicated the need for further study either by repeating the complete robot project on their own or by preparing an adaptation of the workshop for their class.
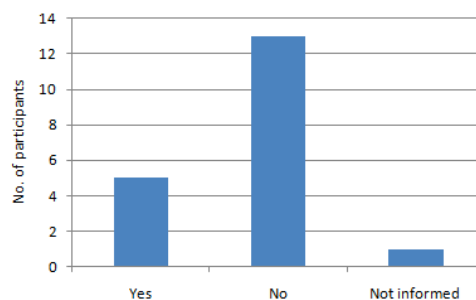
Figure 8. "Do you feel prepared to apply the workshop in your classroom?"

This shows that as result of the workshop the participants achieved the learning objectives on a level of advanced beginner as they recognize that they learned basic computing concepts. However, our results also indicate that such a taster workshop is not sufficient to take participants to higher competence development stages (based on the Dreyfus–Dreyfus model (Dreyfus & Dreyfus, 80), e.g. on the expert or master level, in order to enable them to articulate their expertise and to teach computing to others.

6.2.2 Does the workshop facilitate learning?

All participants considered all sections excellent or good (Fig. 9). All participants also indicated that they would recommend the workshop to their colleagues.

Figure 9. General quality of the workshop

All participants also found section 1 of the workshop easy or very easy and had fun during the workshop. The content of the complete workshop (section 1/2/3) has been considered excellent/good as well as the teaching material. Participants also considered the instructional hands-on strategy adopted in section 1 either excellent (8 participants) or good (11 participants). The complex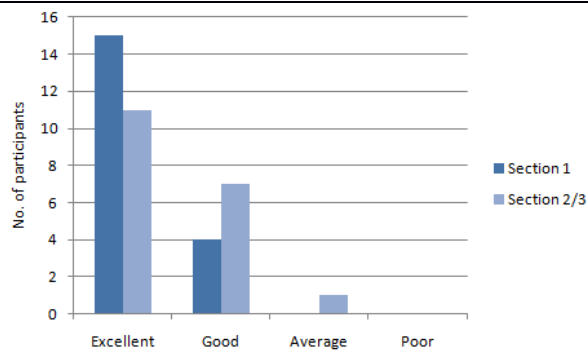ity of section 3 of the workshop presenting the preparation of the technical infrastructure has also been considered adequate by most, with only one exception considering the content too advanced. They also liked the robot-themed project used in the workshop, considered excellent by 3 and good by 16 participants.

Table 5. Comments on strengths and weaknesses of the workshop

Strengths
The ability to work complex concepts in an interactive way, testing and learning.
Scratch programming.
The technical part, the preparation of the workshop material.
The form in which the information was presented.
Experimentation, contact with the software and their operation/installation.
Understanding the logic of programming and the practical part.
Everything was important, mentally visualizing the application within the discipline.
Knowing a little more about Scratch, understanding a little more, and the communication with the robot.
I liked the ideas, the pedagogical possibilities.
Everything.
Possibility of expanding students' knowledge with the programming language.
The frank talk and the "bottom-up" approach.
I liked everything, it was very important to attend the workshop.
The practical part, installations, coding and etc.
Knowing that there is a possibility to apply this at school. The workshop was a contact that will allow to further extend this knowledge.
Weaknesses
Short duration.
The methodological - pedagogical discussion.
Knowing that the project does not have a broad staff structure to develop the project together with technology teachers in schools.
Due to the short duration, I was left with the feeling that I lacked to practice what was explained.
The wish to apply but not having all the materials and my colleague not coming to the workshop.
More "hands-on" and less lecture.
Should have a duration of more days.

Several participants suggested a longer duration with more time to freely explore and apply the learned competencies (posing challenges to be met by the participants) and/or a continuation through additional workshops or courses. Several participants also expressed their wish to involve more colleagues of the educational network.

6.2.3 Does the workshop provide a positive perception of computing and motivate teachers to teach computing?

The workshop changed the perception of the teachers as indicated by their comments (Table 6).

Table 6. Discursive comments on changes on perception on computing

Made activities more concrete.

Like something fun, producing something.
Simplified the topic.
Confirmed how I can teach using programming.
It is possible to teach and understand logic.
Made the understanding of programming possible.
It helps seeing the topic being explained.
It made me remembering what I studied during my undergraduate studies.
I did not know that programming could be so intuitive for laypeople.
Took fear away.
Solution to many problems.
It is possible to carry out the unit in the classroom.

All participants also indicated that the workshop motivated them to apply such a workshop at their schools. This motivation could also be observed based on reactions shortly after the workshop. One teacher even re-built a wooden model of the robot (due to the fact that the school where he works has no access to ATTO material) in order to apply the workshop in his classes (Figure 10).
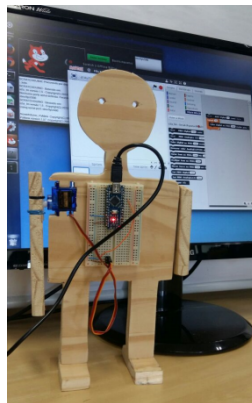

Figure 10. Wooden robot model

Several teachers also got in contact after the workshop with questions, indicating that they were trying to adopt the workshop in their classes.

6.3 Threats to Validity

Several factors in the research design of our study may have influenced the validity of the results. One threat is related to the way of measuring the evaluation objectives. To reduce errors, we adopted a systematic measurement approach (Basili et al., 94) to refine the evaluation objective in analysis questions and measures, operationalized by data collection instruments. In order to reduce threats due to misunderstandings, the questionnaires were carefully designed, reviewed and piloted using the target audience's language. Answers have also been compared to the observations and informal comments collected during the workshop.

Another threat may be the sample size. However, a sample size of only 19 teachers is acceptable for an initial explorative study in order to receive a first feedback. To allow generalization of the results, it will be necessary to repeat the study with larger and diversified samples. Still, the results of this study are a first significant feedback on the application of the instructional unit in the context of an exploratory research.

7. Conclusion

Following the trend for computing education in middle schools, this article proposes a way on how to motivate and train in-service teachers of diverse knowledge areas as a first step in an ongoing comprehensive support. Preliminary results indicate that our taster workshop on physical computing is effective in that it has motivated the majority of participants to introduce computing into their classes. The workshop also provides teachers with a first understanding on computing/programming knowledge, technological knowledge as well as possibilities on

how to integrate the workshop in a multidisciplinary way in their classes. However, our results also highlight that in order to enable teacher to apply the workshops effectively longer training courses as well as ongoing support are required.

References

Alimisis, D.; Frangou, S.; Papanicolaou, K. (2009). *A Constructivist Methodology for Teacher Training in Educational Robotics: the TERECoP Course in Greece through Trainees' Eyes*. Proc. of the 9th IEEE International Conference on Advanced Learning Technologies, Riga, Latvia.

Basili, V. R., Caldeira, G., Rombach, H. D. (1994). *Goal Question Metric Paradigm*. In Encyclopedia of Software Engineering, John Wiley and Sons.

Blum, L. & Cortina, T. J., (2007). *CS4HS: An Outreach Program for High School CS Teachers*. Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, Covington, KY, USA.

Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. Springer.

Bruckman, A. et al. (2009). *Georgia Computes! Improving the Computing Education Pipeline.* Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education, Chattanooga, TN, USA.

Cooper, S., Dann, W., and Harrison, J. (2010). *A K-12 College Partnership*. Proceedings of the 41st ACM Technical Symposium on Computer Science Education, Milwaukee, WI, USA.

CSTA (2017). *CSTA K–12 Computer Science Framework*. The CSTA Standards Task Force, ACM, New York/USA. Retrieved from: https://k12cs.org/.

CSTA (2011). *CSTA K–12 Computer Science Standards – Revised 2011*. The CSTA Standards Task Force, ACM, New York/USA.

Dreyfus, S. E.; Dreyfus, H. L. (1980). *A Five-Stage Model of the Mental Activities Involved in Directed Skill Acquisition*. Washington, DC: Storming Media.

Gal-Ezer, J.; Stephenson, C. (2010). *Computer science teacher preparation is critical*. ACM Inroads, 1(1), 61-66.

Giannakos, M. N. et al. (2014). *Examining and mapping CS teachers' technological, pedagogical and content knowledge (TPACK) in K-12 schools*. Proc. of Frontiers in Education Conference, Madrid, Spain.

Google & Gallup. (2015). *Searching for Computer Science: Access and Barriers in U.S. K-12 Education*. Retrieved from: http://services.google.com/fh/files/misc/searching-for-computer-science_report.pdf.

Goode, J. (2008). *Increasing Diversity in K-12 Computer Science: Strategies from the Field*. Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, Portland, OR, USA.

Gresse von Wangenheim, C.; von Wangenheim, A.; Pacheco, F. S.; Hauck, J. C. R.; Ferreira, M. N. F. (2017). *Teaching Physical Computing in Family Workshops*. ACM Inroads, 8(1), 48-51.

Harvey, B. & Mönig, J. (2017). *Snap! (Build Your Own Blocks) 4.0*. Retrieved from: http://snap.berkeley.edu/.

Jackson, L.; Lamar, C.; Brown, Q.; Latson, V. (2014). *Introducing the Big Ideas of Computer Science through a K-12 Teacher Professional Development Workshop*. Proc. of the Middle Atlantic Section of American Society for Engineering Education, Indianapolis, IN, USA.

Kim, H.; Choi, H.; Han, J.; So, H.-J. (2012) *Enhancing teachers' ICT capacity for the 21st century learning environment: Three cases of teacher education in Korea*. Australasian Journal of Educational Technology, 28(6), 965-982.

Lauwers, T.; Hamner, E.; Nourbakhsh, I. (2010). *A strategy for collaborative outreach: lessons from the CSbots project*. Proceedings of the 41st ACM technical symposium on Computer science education, Milwaukee, Wisconsin, USA.

Liu, J.; Wilson, J.; Hemmenway, D.; Xu, Y.; Lin, C.-H. (2015) *Oh SNAP! A One-Week Summer Computing Workshop for K-12 Teachers*. Proc. of the 10th International Conference on Computer Science & Education, Cambridge University, UK.

Liu, J.; Hassan, E. P.; Barnett, Z. D., Zhang, P. (2011). *A Survey on Computer Science K-12 Outreach: Teacher Training Programs*. 41st ASEE/IEEE Frontiers in Education Conference, Rapid City, SD, USA.

Liu, C. Lin, E. Hasson and Z. Barnett (2011). *Introducing Computer Science to K-12 through a Summer Computing Workshop for Teachers*. Proceedings of the 42nd SIGCSE Technical Symposium on Computer Science Education, Dallas, TX, USA.

S. Y. 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 Behaviour, vol. 41, pp. 51–61.

P. Mishra, P. & Koehler, M.J. (2006). *Technological Pedagogical Content Knowledge: A new framework for teacher knowledge*. Teachers College Record, 108(6), 1017–1054.

MIT. (2016). *Scratch*. http://scratch.mit.edu/.

Naughton, J. (2012). *Why All Our Kids Should Be Taught How to Code*. The Guardian. Guardian News and Media.

Ni, L., Guzdial, M., Tew, A.E., Morrison, B., and Galanos, R. (2011). *Building a Community to Support HS CS Teachers: the Disciplinary Commons for Computing Educators*. Proceedings of the 42st ACM Technical Symposium on Computer Science Education, Dallas, TX, USA.

O'Sullivan, D. & Igoe, T. (2004). *Physical Computing – Sensing and Controlling the Physical World with Computers*, Boston. MA: Thomson Course Technology.

Partanen, T.; Mannila, L.; Poranen, T. (2016) *Learning programming online: A Racket-course for elementary school teachers in Finland*. Proc. of the 16th Koli Calling International Conference on Computing Education Research, Koli, Finland.

Rodger, S. H. et al. (2009) *Engaging middle school teachers and students with alice in a diverse set of subjects*. Proc. of the 40th ACM Technical Symposium on Computer science education, Chattanooga, TN, USA.

Ruzzenente, M., Koo, M., Nielsen, K., Grespan, L. & P. Fiorini, P. (2012). *A review of robotics kits for tertiary education*. Proc. of 3rd Intl. Workshop on Teaching Robotics & Teaching with Robotics, Riva del Garda, Italy.

Shulman, L. (2000) *Teacher Development: Roles of Domain Expertise and Pedagogical Knowledge*. Journal of Applied Developmental Psychology, 21(1), 129-135.

Wohlin, C. et al. (2012). *Experimentation in Software Engineering*. Springer Verlag.

Yin, R. K. (2013). *Case Study Research: Design and Methods*. SAGE Publications, Inc; 5. Edition.

Yorinks, A. (2013). *MrYsLab/s2a_fm*. https://github.com/MrYsLab/s2a_fm.