Advances in Intelligent Systems and Computing 457

Munir Merdan
Wilfried Lepuschitz
Gottfried Koppensteiner
Richard Balogh *Editors*

Robotics in Education

Research and Practices for Robotics in STEM Education

Springer

Advances in Intelligent Systems and Computing

Volume 457

Series editor

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Munir Merdan · Wilfried Lepuschitz
Gottfried Koppensteiner · Richard Balogh
Editors

Robotics in Education

Research and Practices for Robotics in STEM Education

Springer

Editors
Munir Merdan
Practical Robotics Institute Austria (PRIA)
Vienna
Austria

Gottfried Koppensteiner
Practical Robotics Institute Austria (PRIA)
Vienna
Austria

Wilfried Lepuschitz
Practical Robotics Institute Austria (PRIA)
Vienna
Austria

Richard Balogh
URPI FEI STU
Bratislava
Slovakia

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-42974-8 ISBN 978-3-319-42975-5 (eBook)
DOI 10.1007/978-3-319-42975-5

Library of Congress Control Number: 2016946927

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

We are glad to present the proceedings of the 7th International Conference on Robotics in Education (RiE) held in Vienna, Austria, during April 14–15, 2016. The RiE is organized every year with the goal to provide researchers in the field of Educational Robotics the opportunity for the presentation of relevant novel researches in a strongly multidisciplinary context.

Educational Robotics is an innovative way for increasing the attractiveness of science education and scientific careers in the view of young people. Robotics represents a multidisciplinary and highly innovative domain encompassing physics, mathematics, informatics and even industrial design as well as social sciences. As a multidisciplinary field, it promotes the development of systems thinking and problem solving. Moreover, due to various application areas, teamwork, creativity and entrepreneurial skills are required for the design, programming and innovative exploitation of robots and robotic services. Robotics confronts learners with the four areas of Science, Technology, Engineering and Mathematics (STEM) through the design, creation and programming of tangible artifacts for creating personally meaningful objects and addressing real-world societal needs. As a consequence, it is regarded as very beneficial if engineering schools and university program studies include the teaching of both theoretical and practical knowledge on robotics. In this context current curricula need to be improved and new didactic approaches for an innovative education need to be developed for improving the STEM skills among young people. Moreover, an exploration of the multidisciplinary potential of robotics towards an innovative learning approach is required for fostering the pupils' and students' creativity leading to collaborative entrepreneurial, industrial and research careers in STEM.

In these proceedings we present the latest achievements in research and development in educational robotics. The book offers a range of methodologies for teaching robotics and presents various educational robotics curricula and activities. It includes dedicated chapters for the design and analysis of learning environments as well as evaluation means for measuring the impact of robotics on the students' learning success. Moreover, the book presents interesting programming approaches as well as new applications, the latest tools, systems and components for using robotics. The presented applications cover the whole educative range, from elementary school to high school, college, university and beyond, for continuing education and possibly outreach and workforce development. The book provides a framework involving two complementary kinds of contributions: on the one hand on technical aspects and on the other hand on didactic matters. In total, 25 papers are part of these proceedings after careful revision. We would like to express our thanks to all authors who submitted papers to RiE 2016, and our congratulations to those whose papers were accepted.

This publication would not have been possible without the support of the RiE International Program Committee and the Conference Co-Chairs. The editors also wish to express their gratitude to the volunteer students and local staff, which significantly contributed to the success of the event. All of them deserve many thanks for having helped to attain the goal of providing a balanced event with a high level of scientific exchange and a pleasant environment. We acknowledge the use of the EasyChair conference system for the paper submission and review process. We would also like to thank Dr. Thomas Ditzinger and Springer for providing continuous assistance and advice whenever needed.

Vienna, Austria	Munir Merdan
Vienna, Austria	Wilfried Lepuschitz
Vienna, Austria	Gottfried Koppensteiner
Bratislava, Slovakia	Richard Balogh

Organization of RiE 2016

Co-Chairpersons

Richard Balogh, Slovak University of Technology in Bratislava, SK
Wilfried Lepuschitz, Practical Robotics Institute Austria, AT
David Obdržálek, Charles University in Prague, CZ

International Program Committee

Dimitris Alimisis, Edumotiva-European Lab for Educational Technology, GR
Julian Angel-Fernandez, Vienna University of Technology, AT
Jenny Carter, De Montfort University in Leicester, GB
Dave Catlin, Valiant Technology, GB
Stavros Demetriadis, Aristotle University of Thessaloniki, GR
G. Barbara Demo, DipartimentoInformatica—Universita Torino, IT
Jean-Daniel Dessimoz, Western Switzerland University of Applied Sciences and Arts, CH
NikleiaEteokleous, Robotics Academy—Frederick University Cyprus, CY
Hugo Ferreira, Instituto Superior de Engenharia do Porto, PT
Paolo Fiorini, University of Verona, IT
Carina Girvan, Cardiff University, GB
GrzegorzGranosik, Lodz University of Technology, PL
IvayloGueorguiev, European Software Institute Center Eastern Europe, BG
Martin Kandlhofer, Graz University of Technology, AT
BoualemKazed, University of Blida, DZ
Gottfried Koppensteiner, Practical Robotics Institute Austria, AT
TomášKrajník, University of Lincoln, UK
Miroslav Kulich, Czech Technical University in Prague, CZ
Chronis Kynigos, University of Athens, GR

Lara Lammer, Vienna University of Technology, AT
Martin Mellado, Instituto ai2—UniversitatPolitècnica de València, ES
Munir Merdan, Practical Robotics Institute Austria, AT
Michele Moro, University of Padova, IT
MargusPedaste, University of Tartu, EE
Pavel Petrovič, Comenius University in Bratislava, SK
Alfredo Pina, Public University of Navarra, ES
Pericle Salvini, BioRobotics Institute—ScuolaSuperioreSant'Anna, IT
João Machado Santos, University of Lincoln, GB
Alexander Schlaefer, Hamburg University of Technology, DE
Fritz Schmöllebeck, University of Applied SciencedTechnikum Wien, AT
FrantišekŠolc, Brno University of Technology, CZ
Gerald Steinbauer, Graz University of Technology, AT
Roland Stelzer, INNOC—Austrian Society for Innovative Computer Sciences, AT
DavorSvetinovic, Masdar Institute of Science and Technology, AE
Igor M. Verner, Technion—Israel Institute of Technology, IL
Markus Vincze, Vienna University of Technology, AT
Francis Wyffels, Ghent University, BE

Local Conference Organization

Gottfried Koppensteiner, Vienna Institute of Technology/Practical Robotics Institute Austria, AT
Wilfried Lepuschitz, Practical Robotics Institute Austria, AT
Munir Merdan, Practical Robotics Institute Austria, AT

Contents

Part I Didactic and Methodologies for Teaching Robotics

Part II Educational Robotics Curricula

Part III Design and Analysis of Learning Environments

Part IV Technologies for Educational Robotics

Part V Measuring the Impact of Robotics on Students' Learning

Part I
Didactic and Methodologies for Teaching Robotics

Activity Plan Template: A Mediating Tool for Supporting Learning Design with Robotics

Nikoleta Yiannoutsou, Sofia Nikitopoulou, Chronis Kynigos, Ivaylo Gueorguiev and Julian Angel Fernandez

Abstract Although the educational use of robotics is recognised since several decades, only recently they started being broadly used in education, formal and non formal. In this context many different technologies have emerged accompanied by relevant learning material and resources. Our observation is that the vast number of learning activities is driven by multiple “personal pedagogies” which results in the fragmentation of the domain. To address this problem we propose the construct of “activity plan template”, a generic design tool that will facilitate different stakeholders (teachers, instructors, researchers) to design learning activities for different robotic toolkits. In this paper we discuss the characteristics of the activity plan template and the research process employed to generate it. Since we report work in progress, we present here the first version of the activity plan template, the construction of which is based on a set of best practices identified and on previous work for the introduction of digital technologies in education.

Keywords Activity plan template · Learning design · Educational robotics

---

N. Yiannoutsou (✉) · S. Nikitopoulou · C. Kynigos
UoA ETL, National and Kapodistrian University of Athens, Athens, Greece
e-mail: nyiannoutsou@ppp.uoa.gr

S. Nikitopoulou
e-mail: sophieniki@ppp.uoa.gr

C. Kynigos
e-mail: kynigos@ppp.uoa.gr

I. Gueorguiev
ESI CEE, European Software Institute Center Eastern Europe, Sofia, Bulgaria
e-mail: ivo@esicenter.bg

J.A. Fernandez
ESI CEE ACIN Institute of Automation and Control,
Vienna University of Technology, Vienna, Austria
e-mail: julian.angel.fernandez@tuwien.ac.at

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_1

1 Introduction

The educational robotics landscape is vast and fragmented in and outside schools. In the last two decades, robots have started their incursion into the formal educational system. Although diverse researchers have stressed the learning potential of robotics, the slow pace of their introduction is partially justified by the cost of the kits and the schools' different priorities in accessing technology. Recently, the cost of kits has decreased, whereas their capabilities and the availability of supporting hardware and software has increased [1, 2]. With these benefits, educational robotics kits have become more appealing to schools. In this context, various stakeholders—technology providers, teachers, academics, companies focusing on delivering educational material etc.—invest in the creation of different learning activities around robotic kits, in order to showcase their characteristics and make them attractive in and out of schools. Thus, a growing number of learning activities have emerged. These activities share common elements but they are also very diverse in that they address different aspects of Robotics as teaching and learning technology with their success lying in how well they have identified these aspects and how well they address them. This is partly due to the fact that Robotics is a technology with special characteristics when compared to other learning technologies: they are inherently multidisciplinary, which in terms of designing a learning activity might mean collaboration and immersion into different subject matters; they are extensively used in settings of formal and non formal learning and thus involving different stakeholders; their tangible dimension causes perturbations —especially in formal educational settings—which are closely related to the introduction of innovations in organizations and schools (i.e. from considering classroom orchestrations to establishing or not, connections with the curriculum etc.); they are at the heart of constructionist philosophy for teaching and learning [3]; they are relevant to new learning practices flourishing now over the internet like the maker movement, "Do It Yourself" and "Do It With Others" communities etc. With this in mind, we argue that we need to take a step back from the level of specific learning activities and create a more generic design instrument i.e. an activity plan template, which: (a) it will be pedagogically grounded on the particular characteristics of robotics as a teaching and learning tool (b) it will be adaptable to different learning settings (formal–non-formal) (c) it will afford generating different examples of learning activities for different types of kits (d) it will focus on making explicit the implicit aspects of the learning environment and (e) it will urge designers to think "out of the box" by reflecting its content. In the following sections we describe the theoretical background supporting the concept of activity plan template as a design instrument and the method for developing an activity plan template for teaching and learning with Robotics.

2 Theoretical Background

Aiming to explain, in this section, the role of a generic design instrument such as the activity plan template, in addressing the problem of fragmentation in the practice of using educational robotics for learning, we will discuss the dimensions and functions of design in education.

Everyone designs who devises courses of action aimed at changing existing situations into desired ones [4, cited in 5]. With this definition we aim to highlight that design is an integral part of the teaching profession. Acknowledging this dimension in teaching, and with the advent of digital technologies in schools, design based research has been implemented as an approach to orchestrate and study the introduction of innovation in education [6]. Furthermore, in the field of education, design has been introduced as the bridge between theory and practice [5] because design is expected to play a dual role: (a) to guide practice informed by theory and (b) to inform back the theory after the evaluation of the design in practice. Thus, in this context, design is not only an organized sequence of stages, all of which compose an orchestration of the learning process [7] but it is also a reflection and an evaluation tool.

Gueudet and Trouche [8] focusing mainly on resources and documents designed by teachers (e.g. activity or lesson plans), reveal another dimension of design as they describe it as a tool that not only expresses but also shapes the teacher's personal pedagogies, theories, beliefs, knowledge, reflections and practice. The term they use to describe this process is Documentational Genesis. A core element of this approach is instrumental theory [9] according to which the characteristics of the resources teachers select to use, shape their practice on the one hand (instrumentation) and on the other hand, the teachers' knowledge shapes the use of the resources as teachers appropriate them to fit their personal pedagogies (instrumentalization). As a result of the above, teacher designs, according to Pepin et al. [10], are evolving or living documents—in the sense that they are continuously renewed, changed and adapted.

Design as expressive medium for teachers and educators, can also function as an instrument for sharing, communicating, negotiating and expanding ideas within interdisciplinary environments. This property of teacher designs is linked to the concept of boundary objects and boundary crossing [11]. The focus here is on the artefact (in our case activity plan) that mediates a co-design process by helping members of different disciplines to gain understanding of each other's perspectives and knowledge. Educational Robotics for STE(A)M is such an interdisciplinary environment which involves an understanding of related but different domains (i.e. Science, Technology, Engineering, Arts, Mathematics) and involves players from industry, academia and organizers of educational activities.

A problem with all these designs, especially when they involve integration of technologies, is that they are driven by a multitude of "personal pedagogies" the restrictions of which result in adapting technologies to existing practices [12]. Conole (ibid) argues that the gap between the potential of digital technologies to support learning and their implementation in practice can be bridged with a "mediating artefact" to support teacher designs. She continues claiming that such a mediating artefact should be structured according to specific pedagogic approaches and should focus on abstracting essential and transferable properties of learning activities that are not context bound. The activity plan template can play the role of the mediating artefact equipping professionals with a structured means to describe, share and shape their practices. This way we can contribute in addressing the problem of fragmentation in the learning activities regarding the use of Educational robotics.

3 Developing an Activity Plan Template for Educational Robotics

The work reported in this paper takes place in the context of the European project ER4STEM. The main objective of this project is to refine, unify and enhance current European approaches to STEM education through robotics in one open operational and conceptual framework. The development of activity plan templates contributes towards this direction as it provides a generic design instrument that identifies critical elements of teaching and learning with robotics based in theory and practice and in that contributes to the description of effective learning and teaching with robotics. The process through which we develop the activity plan templates in this project includes the following steps: We create a first draft based on (a) on identifying and analyzing a set of good practices and (b) previous work on activity plans that involve innovative use of technologies for teaching and learning. The next step is to use this first draft to design and implement workshops with Robotics in different educational settings and systems. During this implementation we will collect data that will allow us to evaluate, refine and re-design the activity plan template so as to be a useful and pedagogically grounded instrument for designing learning activities. In this paper we are at the first stages of our research and thus we will report on: (a) a set of criteria that we developed in order to identify good practices and (b) the first draft of the activity plan template.

3.1 Identifying Best Practices

The criteria for selecting best practices in the domain of educational robotics were formed through a bottom-up empirical process. Specifically, three researchers from different research teams of the consortium worked independently to select a set of best practices from robotics conferences, competitions, seminars and workshops organized by different institutions. This was the first phase of the selection process, which was not done in a structured way. The second phase included analysis and reflection on phase one. Specifically, the criteria were shaped by (a) an analysis of the content of five examples of best practices already selected and (b) elaboration of the criteria that researchers had implicitly applied during the selection of the specific best practices. Next the items that—from the analytic and the reflective process—were identified to be part of what could be considered best practice in the field of educational robotics were synthesized in one document.

The best practice selection criteria are designed to feed into the activity plans (and not map directly into them) by providing interesting and new ideas for (a) concepts, objectives, artefacts (b) orchestration (c) teaching interventions and learning process (d) implementation process and (e) evaluation process.

Criteria.

The criteria developed for identifying best practices are divided in two categories. One category is mainly a set of prerequisites, which should be covered in order for an event or activity to be considered. The other category consists of the main criteria that identify best practice aspects of the activities.

Prerequisites:

- The topic includes concepts related to the following subjects: Science-Technology-Business-Engineering-Art-Mathematics or something from another discipline but related to robotics.
- The activity–event shows that it has constructionist elements: i.e. it is not just a presentation of tools or predefined guidelines.
- The activity–event is innovative, related to student or citizen interests.
- The activity–event includes technology related to educational robotics.

Main Criteria.

In case that the "educational robotic event" is assessed as relevant according to the aforementioned basic pre-requisites, then the process continues with the assessment of the following parameters (see Table 1). Not all parameters have to be met in order for an event or activity to be considered as good practice. On the contrary, these parameters help us to collect good practices with respect to different dimensions of robotics activities stemming from different sources.

3.2 *First Version of the Activity Plan Template*

In this section we discuss the rationale and the main structure of the first version of the activity plan template. The basic pedagogical theory underlying its design is constructionism, where learning is connected to powerful ideas inherent in constructions with personal meaning for the students. Another aspect underlying our design rationale is the emphasis on the social dimension of the construction process aiming to cultivate a specific learning attitude growing out of sharing, discussing and negotiating ideas. Furthermore, this first version of the activity plan template, is designed to be adaptable to different learning settings (: i.e. formal–non formal),

Table 1 Criteria used for the selection of best practices

Parameters	Description
Context	● **Place**: provides information about the space where the educational robotic activity takes place. This information is crucial to determine other aspects of the learning design such as orchestration issues, formal or non-formal settings etc. Possible examples can be school, museum, science institutions, or other educational scientific organizations. ● **Participants' description**: provides information regarding issues such as age, number, culture, background etc. The activity is considered as good practice if it is aligned to the age of the participants, the number, the prior knowledge of the participants on a specific subject, etc. ● **Theoretical framework**: refers to the pedagogical approach used in implementing the educational activity e.g. DIY (Do It Yourself), DIWO (Do It With Others), Constructionism, STEM education, Design. In several cases the theoretical framework is implicit and can be inferred from the way the activity is orchestrated and designed.
Educational activity	● **Connection with a curriculum**: This dimension provides information regarding issues of connecting the teaching of robotics to specific topics of national curricula. It is not expected to apply to all events or activities identified. ● **Motivation for the activity**: Provides information on what has motivated the organization of the specific activity (e.g. introducing girls to robotics, elaborating on specific STEM concepts, using art to explore robotics etc.). In identifying good practices we are looking for interesting motivations and the way the activity is organized to support this motivation. Special focus is given to events that are designed to motivate young people to learn STEM disciplines. ● **Description of the activity**: Provides information regarding the implementation of the activity. This information helps out in identifying if the activity matches the context the motivation etc. The activity description is expected to refer to issues regarding the duration, tasks, orchestration, grouping, learner interaction (i.e. where is the emphasis concerning the action, the relationships, the roles in the group and the teacher's role).
Tools	● **Technology used—selection criteria**: Provides information on the specific technology used for the implementation of the activity. It is considered as good practice if the educational robotic event is based on technology that follows the latest trends, it is compatible with the background of the participants, facilitates well the objectives and the motivation of the activity, it is presented in a way that it is understandable by the specific target group in the workshops and is similar to technologies used by young people in their everyday life e.g. mobile and cloud solutions ● **Type of artefacts produced**: This parameter involves the output of the activity or the event. It is considered as good practice if the artefacts produced during the educational robotic event are interesting and engaging; participants are interested to use the artefacts and to apply them in different domains of their lives.
Evaluation	The description of the activity provides information regarding methods and results of its evaluation, including the perspectives of the participants and the reflection of the teacher-instructor on aspects that might need improvement or are going to be changed in next implementations

(continued)

Table 1 (continued)

Parameters	Description
Sustainability	• **Cost of the activity**: This dimension involves information regarding mainly costs of the material and organizational costs. It is considered a good practice if the activity requires materials or tools that are reasonably priced compared to other related activities. • **Activity Financing**: The activity−event is considered a good practice with respect to this dimension if it has a sustainable model for financing in mid-term period, e.g. self-financing through fees, wide voluntary base, partnership with public organizations such as municipalities, schools or long term sponsorship partners. • **Activity Repetition**: An activity−event is considered a good practice if it is performed sustainably for at least three subsequent periods in close cooperation with schools or other educational organizations.
Accessibility	The information regarding this parameter involves mainly the sharing of activity related material (i.e. manuals, guidelines etc.), in a way (i.e. open access, structuring of information) that allows the activity−event to be replicated by other relevant stakeholders.

thus, its structure is modular and the intention is to allow “selective exposure” of its elements to different stakeholders (the term “selective exposure” is borrowed from Blikstein [13] to describe the intentional hiding of some of the template elements, according to the relevant settings or stakeholders).

This first version discussed here, is informed by an analysis of the best practices identified and it is based on previous work on activity plan templates that aim at the integration of digital technologies in learning [14]. The structure of the Activity plan template is presented in detail in Table 2 and addresses the following aspects: (a) the description of the scenario with reference to the different domains involved, different types of objectives, duration and necessary material; (b) contextual information regarding space and characteristics of the participants; (c) social orchestration of the activity (i.e. group or individual work, formulation of groups etc.); (d) a description of the teaching and learning procedures where the influence of the pedagogical theory is mostly demonstrated; (e) expected student constructions; (f) description of the sequencing and the focus of activities; (g) means of evaluation.

Future work will focus on refinement of the activity plan template through its use by ER4STEM partners to create their activity plans and through data collected during the implementation of these activity plans in realistic situations (workshops).

Table 2 First version of the activity plan template

Title	
Author	Teacher, Designer
1. Focus, set up and requirements for the activity	
Domain	• **Primary domain** (Select one of the following): Science; Technology; Business; Engineering; Arts; Mathematics. • **Contextual (Peripheral) domain** (provide a rating of the level of emphasis on concepts in each of these domains): Science (0–10); Technology (0–10); Business (0–10); Engineering (0–10); Arts (0–10); Mathematics (0–10).
Objectives	Objectives are organized in a set of four different categories: • **Subject matter**: i.e. study the angle and position of all materials (servo motors, circuits, sensors), as well as the construction of the legs in order for the robot—insect to be autonomous and move correctly. • **Technology use**: i.e. Programming with Arduino. • **Social and collaborative skills**: i.e. develop collaborative skills, take roles within groups. • **Argumentation and fostering of maker culture**: i.e. practice making conjectures about how the robot will react to external stimuli based on the program given.
Time	• Duration: i.e. 5 weeks • Schedule: i.e. 2 h per week
Materials and artifacts	• **Digital artifact**: e.g. programming language, visual interface, robot simulation etc. • **Robotic artifact**: i.e. the technology and the robot form, e.g. an insect, a car etc. • **Student's workbook and manual**: i.e. a manual with step-by-step instructions for the electronic and the programming part. • **Teacher's instruction book and manual**: teacher's notes with a template of e.g. three incisive stages and five steps for the first two stages.
2. Space and students	
Students (target audience):	• **Sex and age**: e.g. boys and girls, 17 years old • **Prior knowledge**: e.g. little if any knowledge of Arduino, good knowledge of electronics • **Nationality and cultural background**: e.g. 5 pupils from Albania and 10 from Greece • **Social status and social environment**: e.g. under-privileged area, mainstream public school, elite private school • **Special needs and abilities**: e.g. ADD, dyslexia, Soc. Em. Behavior Disorders, gifted, other
Space info	• **Organizational and cultural context:** e.g. in school at the technology laboratory, during project time in after school established voluntary club activity. • **Physical characteristics**: e.g. indoors, floor
3. Social orchestration	
Participants	Students: e.g. 15 Tutors: e.g. 2

(continued)

Table 2 (continued)

Title	
Grouping	**Setting**: students in a normal classroom, around light mobile tables, in small groups **Grouping criteria**: mixed ability, mixed gender
Interaction during the activity	**Actions**: exchange ideas, dialogue, negotiation, debate. **Relationships**: collaborative, competitive **Roles in the group**: pre-defined roles, emergent roles **Support by the tutor(s)**: support, intervene, self-regulatory
4. Teaching and learning procedures	
Teacher's role	Mentor, consultant, researcher, instructor
Teaching methods	Demonstrate, engage by example
Student expected activity	Writing, observing, constructing, discussing, negotiating,
Student learning processes	● Designed conflicts and misconceptions: do the activity designers wish to bring students in conflict with mistaken conceptions documented in educational research or their teaching practice? ● Learning processes emphasized: e.g. emphasis on analyzing robot behavior in order to refine and reflect on the code that defines this behavior. ● Expected relevance of alternative knowledge: e.g. students are expected to investigate the structure of an insect's body (biology) in order to construct their robot.
5. Student productions	
Artifacts—robots	● **Assignment**: What tasks shall the robot perform (e.g. entertain, bring things, call help, vacuum clean etc.)? ● **Interaction**: What are the means of communication with the robot (speech, gesture, mind control, buttons, app etc.)? ● **Morphology**: How does the robot look like? What material is it made of (e.g. machine-like, zoomorphic, anthropomorphic, cartoon-like etc.)? ● **Behavior**: What shall the robot behave like (e.g. butler, friend, pet, protector, teacher etc.)? ● **Material**: What parts are needed for the construction of the robot (e.g. electronics, software, mechanics etc.)?
Programming	● Structure of code-commands ● Elements (e.g. iteration, selection, variables) ● Conditionals (e.g. event handling)
Discussion	● **Descriptive—explanatory**: description of a situation, a construct or an idea for others to understand and/or to implement. ● **Alternative**: provision of solutions to problems, provision of alternatives if a dead end is reached. ● **Critical—objection**: revision of other's constructs and ideas, identification of problems, challenge of ideas. ● **Contributory—extending**: sharing of resources, provision of ideas towards improving an existing construct or initial idea.

(continued)

Table 2 (continued)

Title	
6. Sequence and description of activities	
Phasing	• Phase 1: Construction phase—hands on the robot (duration one hour) • Phase 2: Assembly discussion: All groups present the robots they have constructed and discuss challenges and problems (Duration 20 min) • Phase 3: Programming: constructing the robot's behavior. Groups can exchange ideas and ask for help from each other (duration 1 h). • Phase 4: Presentation of the final construct: A short video demonstrating the robot and its behavior or a blog presentation including a photograph, a short description and the code.
7. Assessment procedures	
Formative	• Pupil voice activities (Interviews with students, Questionnaire) • Observation notes • Peer assessment
Summative	• Essays • Tests • Student productions (code-robots-textual discussions) • Mark sheet

4 Conclusion

In this paper we discussed the role of activity plan templates as mediating artifacts in harnessing the potential of educational Robotics for learning and in addressing the issue of fragmentation in the domain. The concept of a mediating artifact was adopted here to describe a generic learning design instrument that is based on: (a) a specific pedagogical theory and (b) the particularities of robotics as technologies. The activity plan template is an abstraction of what we have identified as essential and transferrable elements of learning with robotics. The work reported here is in progress, thus the activity plan template presented, is going to be evaluated in practice by teachers who will use it to create their own activity plans and by researchers and students during the implementation of these plans in practice. Feedback generated from this process will be used to inform the activity plan template so as to achieve (a) a level of abstraction that it will make it adaptable to different settings and (b) a level of detail that will demonstrate the influence of a specific pedagogical approach and will address the particularities of Robotics.

Acknowledgments The project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 665972. Project Educational Robotics for STEM: ER4STEM

References

1. Alimisis, D.: Robotic technologies as vehicles of new ways of thinking, about constructivist teaching and learning: the TERECoP project. IEEE Robot. Autom. Mag. **16**, 21–23 (2009)
2. Miller, M.: Mobile building blocks 2014. Mobile Cores. PC Mag. (2014)
3. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc. (1980)
4. Simon, H.A.: The sciences of the artificial. MIT Press, Cambridge, MA (1969)
5. Mor, Y., Winters, N.: Design approaches in technology-enhanced learning. Interact. Learn. Environ. **15**, 61–75 (2007)
6. The Design-Based Research Collective: Design-based research: an emerging paradigm for educational inquiry. Educ. Res. 5–8 (2003)
7. Trouche, L.: Managing the complexity of human/machine interactions in computerized learning environments: guiding students’ command process through instrumental orchestrations. Int. J. Comput. Math. Learn. **9**, 281–307 (2004)
8. Gueudet, G., Trouche, L.: Towards new documentation systems for mathematics teachers? Educ. Stud. Math. **71**, 199–218 (2009)
9. Verillon, P., Rabardel, P.: Cognition and artifacts: a contribution to the study of though in relation to instrumented activity. Eur. J. Psychol. Educ. **10**, 77–101 (1995)
10. Pepin, B., Gueudet, G., Trouche, L.: Re-sourcing teachers’ work and interactions: a collective perspective on resources, their use and transformation. ZDM Math. Educ. **45**, 929–943 (2013)
11. Kynigos, C., Kalogeria, E.: Boundary crossing through in-service online mathematics teacher education: the case of scenarios and half-baked microworlds. ZDM 1–13 (2012)
12. Conole, G.: The role of mediating artefacts in learning design. In: Handbook of Research on Learning Design and Learning Objects: Issues Applications and Technologies, pp. 108–208 (2008)
13. Blikstein, P.: Computationally enhanced toolkits for children: historical review and a framework for future design. Found. Trends®. Hum.Comput. Interact. **9**, 1–68 (2015)
14. Yiannoutsou, N., Kynigos, C.: Boundary objects in educational design research: designing an intervention for learning how to learn in collectives with technologies that support collaboration and exploratory learning. In: Plomp, T., Nieveen, N. (eds.) Educational Design Research: Introduction and Illustrative Cases, pp. 357–379. SLO, Netherlands Institute for Curriculum Development, Enschede, The Netherlands (2013)

V-REP and LabVIEW in the Service of Education

Marek Gawryszewski, Piotr Kmiecik and Grzegorz Granosik

Abstract The following paper exposes an effective approach of teaching robotics by applying very efficient and popular set of tools. It demonstrates a combination of applications such as V-REP and LabVIEW in order to be later utilized for rapid prototyping, algorithm design and revival of both simple as well as fairly advanced robotic environments. The described teaching methodology has been successfully applied to high-school students during their facultative robotics courses which have been taking place at the Lodz University of Technology. Detailed summary of the results of this exertion is also provided.

Keywords Educational robotics · V-REP · LabVIEW · Simulation environment · Rapid prototyping

1 Introduction

According to various statistics [1, 2], robotics market has been exponentially growing for the last few years and nothing indicates that this could change in the nearest future. Becoming more and more specialized, robots take over a significant number of tasks performed by the human so far. Particular growth can be seen not only in the industrial process automation area, but also, and especially, in consumer robotics.

Even though the prophecies of robots taking over most people's jobs still seem to be a fantasy, the transition is clearly visible.

M. Gawryszewski (✉) · P. Kmiecik · G. Granosik
Institute of Automatic Control, Lodz University of Technology,
B. Stefanowskiego 18/22, 90-924, Lodz, Poland
e-mail: marek.gawryszewski@dokt.p.lodz.pl
URL: http://robotyka.p.lodz.pl

P. Kmiecik
e-mail: piotr.kmiecik@aserio.pl

G. Granosik
e-mail: granosik@p.lodz.pl

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_2

At the moment, this means that we have to get used to the presence of robotic devices in our lives and learn to cooperate with them. Even though some of the job opportunities have already expired, a huge amount of new ones is just opening. However, the transformation we are witnessing must enforce the change in the way we perceive robots and in the direction in which potential workforce should be trained. The conclusion is that robotics should be introduced to the vast majority of students at possibly early level of education—to get them accustomed to the new technologies and in the way that is not overwhelming—to let them develop interest in it.

In the following paragraphs we provide an evaluation of two powerful pieces of software, which when combined together, have a rich potential to significantly facilitate the learning process and may also be very useful in further development. Subsequently, we include a set of instructions of how this can be achieved and eventually we deliver the results of our attempt.

2 Related Work

The demand for having flexible training tools in order to meet the needs of the emerging robotics market and the potential of virtual reality has been recognized quite a long time ago. In [3], KUKA's universal simulation environment, targeted especially for educational purposes, has been presented in contrast to the other, device-specific modelling software available at that time. Another virtual visualization environment, which includes three robotic models and is suitable for teaching robotics at the university level has been presented in [4]. A more contemporary approach, based on Gazebo simulator has been demonstrated in [5]. The summary of the usage of similar tools can be found in [6, 7], where the authors share their experiences gained during the several years of educational activities. The reported results seem to be satisfactory enough to still take on this subject.

3 Motivation and Background

As an interdisciplinary field of science, robotics requires from its adepts to acquire an extremely wide range of knowledge in many differed areas in order to be able to freely express their creativity. It is also worth mentioning, that robotic systems are usually fairly complex, susceptible to damage or even dangerous when handled improperly. They are rather expensive as well, which can often result in a limited access to such devices. Those are only a few reasons why to look for many alternative ways of targeting the challenges thrown by the world of modern robotics. A short description of the tools of our choice is presented as follows.

3.1 V-REP

V-REP *(Virtual Robot Experimentation Platform)* is a versatile, cross-platform, 3D robotic simulation environment made by Coppelia Robotics [8, 9]. It can be used for modelling simple robotic components such as sensors and actuators as well as complex fully functional robotic systems.

The application provides forward and inverse kinematics calculations along with the four real-time physics engines dedicated for dynamic calculations and simulation of full objects interactions.

It supports various interchangeable CAD formats such as URDF, COLLADA, DXF, 3DS, OBJ, STL for easier integration and can be bridged with various external applications through several remote API's (C/C++, Python, Java, Matlab, Octave, Lua, Urbi) and the ROS interface. With its well designed, user friendly and customizable interface it is a very good choice for presenting the basic principles of robotics as well as designing and testing any robotic device.

3.2 LabVIEW

LabVIEW is a widely spread engineering platform from National Instruments [10]. It comes with a visual programming language named *G*, which basically means that the program structure consists of a number of graphical blocks representing individual instructions, all connected with wires to ensure a suitable work-flow. Such an approach guarantees even inexperienced users, unfamiliar with the programming principles to efficiently learn and operate through the environment. The application supports a wide range of data acquisition and instrument control devices which makes it a great tool for working with the real hardware.

4 Course Description

The main objective of the classes is to enable a group of high school students to design, simulate and build a factual mobile robot: a Mars rover. By working on this project, its participants familiarize themselves with a wide range of aspects of the development process: from making the first design choices to the final testing of a working machine.

As a part of the introduction, various instruments were presented to them as the possible ways to solve the whole task: modelling tools, simulators, programming environments, prototype boards and many others. In this article we focus on the two aspects of the whole work: simulations and algorithms development.

One of the most important things in the whole process is the fact, that the course is led by example and try. Students were provided with knowledge of how to begin.

The rest is all about achieving their goals on their own with only a minimal assistance and supervision coming from a teacher.

4.1 The Role of Selected Software

Harnessing V-REP and LabVIEW enables students to benefit from various, well-established tools like inverse kinematics solver right out of the box. This means, that they do not have to understand deeply all aspects of the given problem, e.g. inverse kinematics. Considering less advanced high school attendees this might be crucial to preserve motivation, as some problems might be worked around without decreasing overall complexity of a project.

Up till now, V-REP has been used to demonstrate, how to model and simulate a robot. The application does not only allow to conduct simulations of a designed device, but also of the entire surrounding environment.

During the hands-on sessions, students have been creating models of mobile robots to understand the possibilities and limitations of the whole process. Thanks to built-in modules, they were able to easily simulate dynamics of their creations and plan a path to follow, which included taking into account existing obstacles.

Two methods of implementing robot's algorithms were demonstrated. The first one was scripting in Lua language [11], which is an internal mechanism of V-REP.

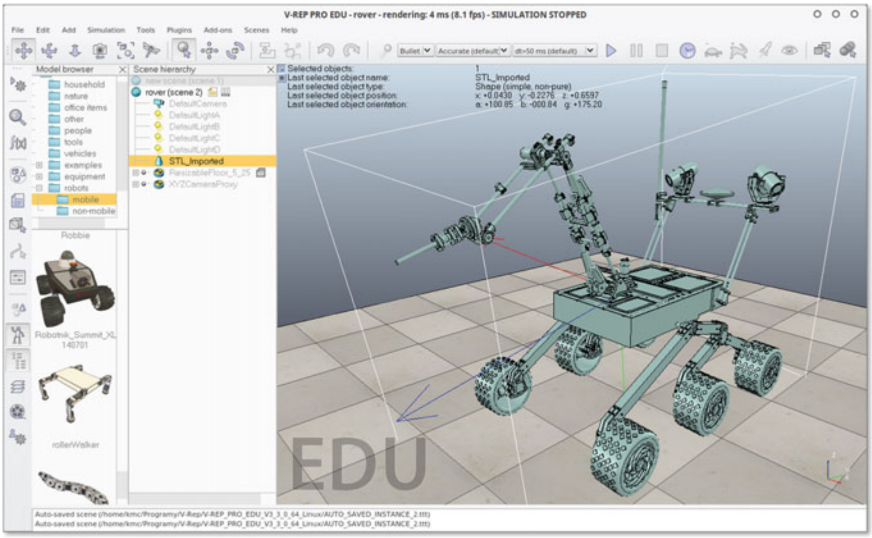
Additionally, the originators of V-REP have established APIs that allow users to create applications of any kind, which then are able to communicate with the simulation and control its elements. Thanks to that, it is now possible to create control algorithm in any modern programming language, but C++, Python, Java, Matlab and LabVIEW are the preferred ones.

As for the second method, we have chosen V-REP's LabVIEW interface, which was originally designed by Peter Mačička [12]. Utilizing his work, we asked the students to create their own programs to control simulated robots.

4.2 Modelling Basics

Modelling with V-REP involves dealing with three basic groups of elements [13]:

1. *Scene objects*—twelve types of construction blocks (e.g. shapes, joints, proximity, force/torque or vision sensors), that can be combined with each other to create more complex designs
2. *Calculation modules*—five basic embedded algorithms: forward and inverse kinematics, path planning, collision detection, physics and dynamics, minimum distance calculation



V-REP PRO EDU - rover - rendering: 4 ms (8.1 fps) - SIMULATION STOPPED

Selected objects: 1
Last selected object name: STL_Imported
Last selected object type: Shape (simple, non-pure)
Last selected object position: x: +0.0430 y: -0.2276 z: +0.6597
Last selected object orientation: a: +100.85 b: -000.84 g: +175.20

```
Auto-saved scene (/home/kmc/Programy/V-Rep/V-REP_PRO_EDU_V3_3_0_64_Linux/AUTO_SAVED_INSTANCE_2.ttt)
Auto-saved scene (/home/kmc/Programy/V-Rep/V-REP_PRO_EDU_V3_3_0_64_Linux/AUTO_SAVED_INSTANCE_2.ttt)
```

Fig. 1 V-REP's default user interface running under Linux operating system (with a model of an authentic Mars rover, built by the students at the Lodz University of Technology [14], imported to the scene)

3. *Control mechanisms*—local and remote interfaces like: embedded scripts, plug-ins, add-ons, remote APIs, ROS nodes and custom (client/server model based) solutions, that can work simultaneously and cooperate with each other

The main program interface (Fig. 1) is clear and simple. By default, it consists of standard toolbars offering the ability to access most of the features. A scene pane which can display multiple views as well as custom user interfaces is available on the right. On the left, there is a browser with thumbnails of all available models that can be dragged into the scene and finally a tree view called *Scene hierarchy*, which represents dependencies between all objects of the project.

When all elements are finally placed on the scene, they can be easily controlled via the internal scripting language or provided remote APIs. To illustrate the functionality more accurately, we can imagine creating a rotary joint on the scene and then setting its velocity by referencing it via the Lua script.

Lua is described as very fast, powerful, yet simple and lightweight scripting language ideal for rapid prototyping. In addition to procedural syntax, it provides meta-mechanisms for implementing elements typical for object-oriented programming languages, like classes and inheritance [15].

There are several types of embedded scripts that can be applied to the project. The most important are those used when the simulation is running which divide into four categories: *Main, Child, Joint Control* and *Contact Callbacks*. Then we can also use the *Customization, General callback* and finally *Dialog* and *Editor* scripts. Below we provide a short example of a *Child* script, which demonstrates referencing objects in V-REP.

Lua script—referencing simulation objects in V-REP

```
-- Handle of the left motor
leftMotor=simGetObjectHandle(``bubbleRob_leftMotor'')
-- Handle of the right motor
rightMotor=simGetObjectHandle(``bubbleRob_rightMotor'')

speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])

if (backUntilTime<simGetSimulationTime()) then
        -- When in forward mode,
        -- move forward at the desired speed
        simSetJointTargetVelocity(leftMotor,speed)
        simSetJointTargetVelocity(rightMotor,speed)
else
        -- When in backward mode,
        -- backup in a curve at reduced speed
        simSetJointTargetVelocity(leftMotor,-speed/2)
        simSetJointTargetVelocity(rightMotor,-speed/8)
end
```

(based on: Line following BubbleRob tutorial)

4.3 Interfacing V-REP with LabVIEW

There are many aspects in which the embedded scripting excels. The ease of the integration, inherent scalability, robustness and compatibility are just only a few of them. However, considering the general purpose of the course, we decided to present the remote APIs as well. Among others, LabVIEW seemed to be the best compromise between the number of available features and intuitiveness.

The interfacing is extremely simple and poses absolutely no problem. On the server side (in our case this means V-REP simulation), we just have to edit the internal script to set the available port, include a proper plug-in called *v_repExtRemoteApi.dll* (the name may vary depending on the platform) and enable remote APIs by calling *simExtRemoteApiStart()*. From the clients perspective, we use a universal *Call Library Function Node* (indicated by arrow in Fig. 2). We just have to make sure, that proper Dynamic-Link Library (*remoteApi.dll*) is set in the block properties and select the adequate parameters (indicated by arrow in Fig. 3). A short tutorial with a wide range of examples has been provided by the author of the interface [12].

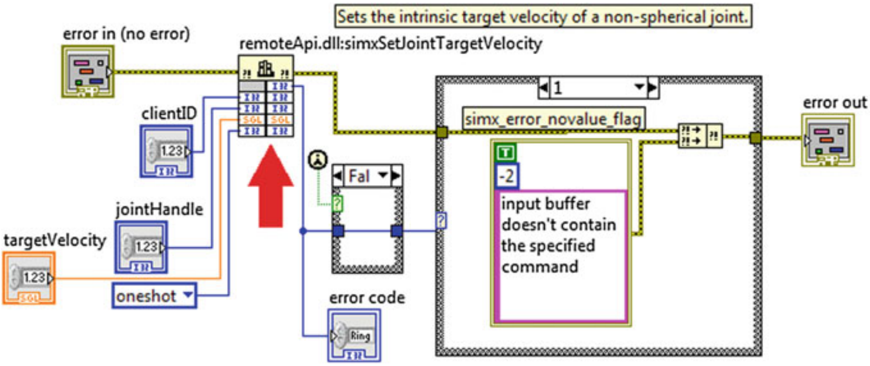
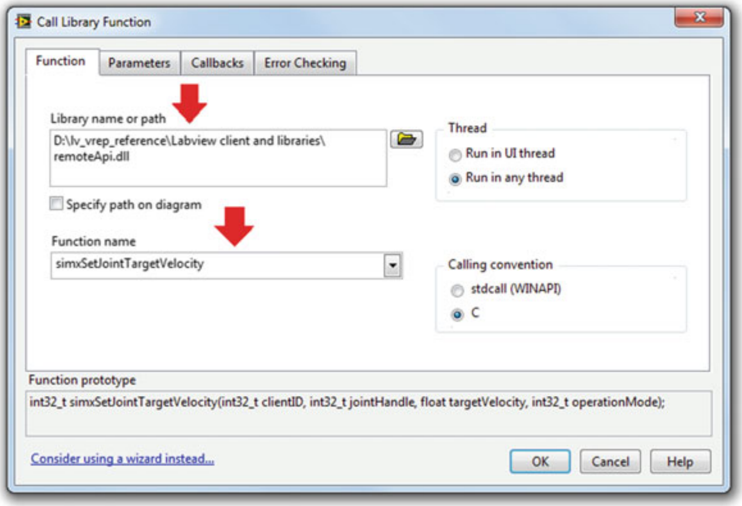


Fig. 2 SubVI handling joint velocity



Library name or path
D:\lv_vrep_reference\Labview client and libraries\remoteApi.dll

☐ Specify path on diagram

Function name
simxSetJointTargetVelocity

Thread
○ Run in UI thread
● Run in any thread

Calling convention
○ stdcall (WINAPI)
● C

Function prototype
int32_t simxSetJointTargetVelocity(int32_t clientID, int32_t jointHandle, float targetVelocity, int32_t operationMode);

Fig. 3 *Call Library Function Node* properties

4.4 Workflow and Educational Values

At the very beginning of the course a modelling software has been presented to students. It is also worth mentioning, that in addition to V-REP, the presentation included Autodesk Inventor, which will later be used to create their own models.

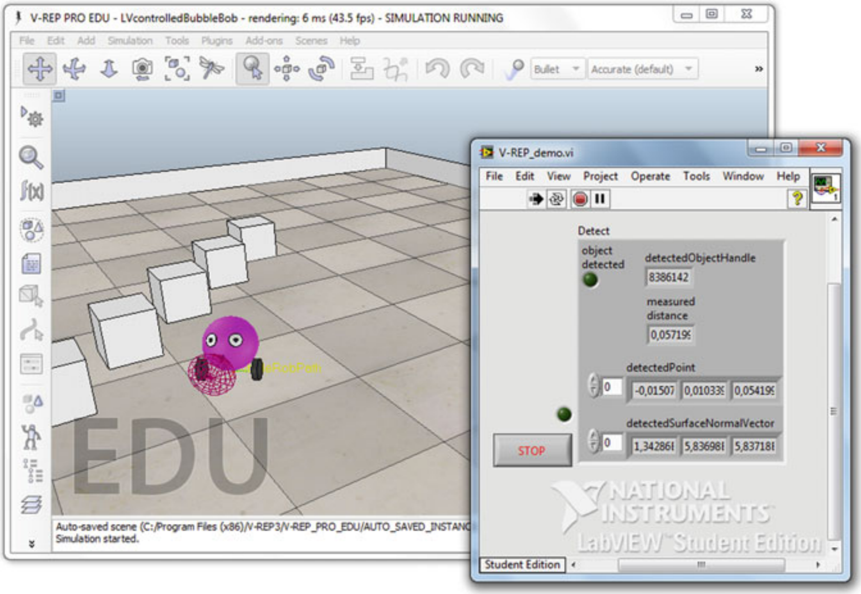
The next step was the introduction to LabVIEW. The lecture included environment basics such as: the distinction between graphs and front panels, basic data types, kinds of controls, indicators and constants as well as execution structures i.e. Loops and Cases. Participants learned how to create simple programs, according to paradigms and patterns used in LabVIEW.

The following four meetings, which are the main scope of this article, were all about modelling in V-REP and programming in LabVIEW.

During the first one, students were shown the capabilities of V-REP, the basic concepts, and how to build simulated objects and elements of the environment. They gained knowledge about its distributed control architecture and the advantages guaranteed by this kind of solution (e.g. portability and scalability). Finally, all of them learned how to associate scripts with individual objects. The possibility of importing prototypes created in the above mentioned Autodesk Inventor has also been demonstrated as shown in Fig. 1.

The second and the third meeting were focused on modelling the BubbleRob robot, which is a part of the official V-REP's tutorial [16]. All the students, divided into the groups of two members each, had to go through the whole process of creating a simulation of a simple mobile robot step-by-step, which definitely allowed them to improve their practical skills. As a result, they were able to see for themselves, how robotic models are being designed and how to resolve typical problems.

During the fourth meeting, two previously introduced tools were connected together: we used LabVIEW to create control application for the robot simulated in V-REP. This has happened in the two stages. The first one was to set up an example control application, which shows how to exchange data in both directions. The difference between all available interfaces has been outlined emphasizing regular and remote APIs. The second one involved students in creating their own programs, to control robot in V-REP (Fig. 4).



Auto-saved scene (C:/Program Files (x86)/V-REP3/V-REP_PRO_EDU/AUTO_SAVED_INSTANC
Simulation started.

Detect

object detected ●

detectedObjectHandle: 8386142

measured distance: 0,05719

detectedPoint: 0 | -0,01507 | 0,01033 | 0,05419

detectedSurfaceNormalVector: 0 | 1,34286 | 5,83698 | 5,83718

Fig. 4 "BubbleRob" robot (*back window*), controlled by LabVIEW application (*front window*)

What is important, as LabVIEW was selected as an environment to create algorithms for the real robot, the students were able to test their programs with the simulation and use it in a real-world application afterwards. They also had a chance to exploit a variety of *Toolkits* available for LabVIEW, including those dedicated especially to robotic applications.

Given the knowledge, that the only thing that differs is a low level communication layer (i.e. function calls) and that it should be replaced by the one appropriate for the selected hardware, they were finally enabled to use the same program with both, the simulated and the actual robot.

The next step of our course is to show the basics of electronics with Arduino [17], which serves as a controller of robots intended for the course. After it is finished, students will be ready to start making real robots, as they gain enough skills to know how to start and they have already seen on simulation that their models can work.

4.5 Survey

A couple of weeks after the last meeting focused on LabVIEW and V-REP, we asked our students to provide feedback about the workshops. Our goal was to understand, the students' opinion about the form of the classes—if they are satisfied with it and the provided content as well as if they noticed any increase in the level of their knowledge.

Eleven participants completed the questionnaiare. There were 9 statements (collected in Table 1) describing the workshop and the responses were formatted in a 5-point Likert scale: answer 1 means: I strongly disagree and 5: I strongly agree.

In general, our work as tutors has been well evaluated, as all the students indicated that tutors were helpful and willing to give explanations if something was unclear. They also agreed that the knowledge was presented in an explicit and understandable manner as shown in Fig. 5, plot: "Questions 1, 2".

On the other hand, the presented content did not fit their needs perfectly, as 4 students marked answer 3 which might be considered as a slightly negative opinion (Fig. 5, plot: "Questions 3, 4").

For the vast majority of the students our workshops were interesting (9 out of 11) and most of them declared to take part in more advanced courses, as an extension of this one (Fig. 6, plot: "Questions 5, 6, 7").

We also asked about the programming skills, and 9 participants claimed that they have improved their skills as a result of the course. Unfortunately, the other two disagreed with this statement.

The question about the further use of the acquired knowledge seems to be unresolved: there are 6 students which claim that they see benefits from the workshop as a way to apply their knowledge in their own projects (Fig. 6, plot: "Question 8, 9"), yet still 5 people has no clear idea or do not see any profit.

Table 1 Questions

No.	Topic	Question
1	Tutors	Tutors were willing to answer questions
2		Knowledge was presented in an explicit and understandable manner
3	Materials	Course materials used were useful
4		Course materials were easy to understand
5	Course form	Course was interesting
6		Course form allows me to take part actively
7		My programming skills raised through participation in the course
8	Future	I would like to take part in more advanced edition of the course in the future
9		Knowledge gained during course will allow me to implement my own ideas

This was the first iteration of workshops and this survey showed us some space for improvements. We indicated three crucial areas which should be reworked before the next edition:

- *the content*—the course was based on step-by-step, written instructions, which were generally considered as useful, but there is some space for refinement such as dividing the tutorial into smaller parts or introducing better balanced level of difficulty.
- *programming skills*—part that introduces LabVIEW shall be reworked to another form.
- *students' ideas*—the main goal of the entire project is to build a Mars rover. It is also very important to create a strong the base to develop students' own ideas on basis of this project, and this needs to be emphasized.

5 Final Remarks

The most important aim of our project was to introduce fairly complex robotic problems to young, inexperienced people and additionally solve these problems in an

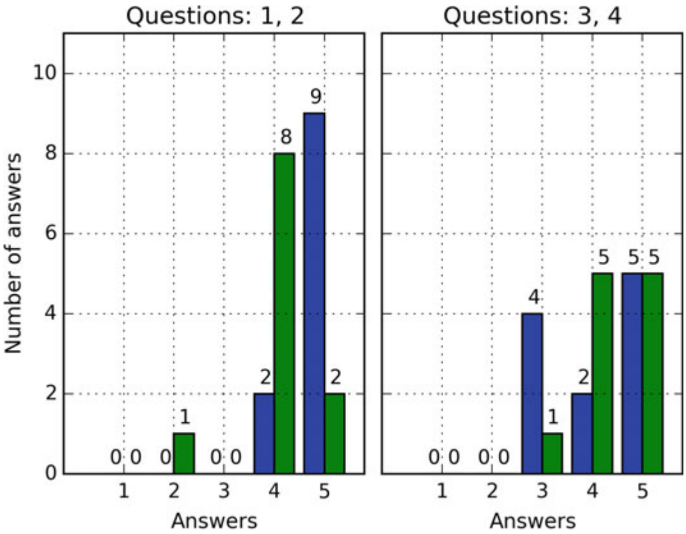


Fig. 5 Histogram of answers the the questions: 1—"Tutors were willing to answer questions." (*left plot*, *blue*), 2—"Knowledge was presented in an explicit and understandable manner." (*left plot*, *green*), 3—"Course materials used were useful." (right plot, blue), 4—"Course materials were easy to understand." (*right plot*, *green*)

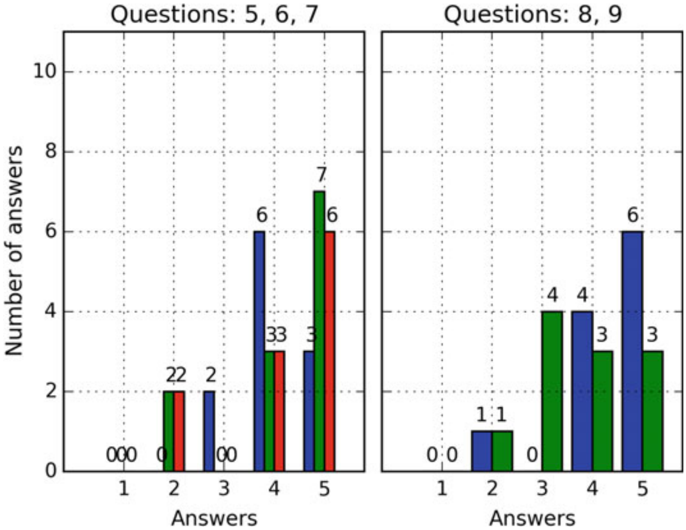


Fig. 6 Histogram of answers the the questions: 5—"Course was interesting." (*left plot*, *blue*), 6—"Course form allows me to take part actively." (*left plot*, *green*), 7—"My programming skills raised through participation in the course." (*left plot*, *red*), 8—"I would like to take part in more advanced edition of the course in the future." (*right plot*, *blue*), 9—"Knowledge gained during course will allow me to implement my own ideas." (*right plot*, *green*)

interesting manner with the available (possibly free) tools. These issues are present in most projects dealing with kids and teenagers [18, 19].

Among several simulation environments like Webots [20] or Gazebo [21] we have chosen V-REP as the most engaging and accessible. There are at least five great premises in favour of this decision:

1. *Usability*—representation of the scene is done in a similar way as in any other modelling software, which students are already familiar with way before they get introduced to V-REP
2. *Efficiency*—the vast majority of work can be done by simply using drag-and-drop functionality which considerably improves productivity
3. *Scalability and portability*—a great variety of ways to add logic to the scene: by internal scripts or external applications over standardized communication channels
4. *Compatibility*—possibility to import/export objects in various CAD formats, so the elements in simulation can be based on ones created in modelling software
5. *Simulation capabilities*—it is important to see how the mass and inertia of the robot influences its ability to move or perform other tasks and V-REP provides four different models of dynamics, all of them available free of charge for educational purposes

Our goal was to deliver tools and methods that are fun, easy to use and can be utilized with limited training, as our final goal is to build a robot, not to learn new programming environments.

We believe, that our approach will finally succeed, as participants of the course are now able to build simulations and control algorithms for their own robots with a very limited supervision.

We also hope, that our experience will be helpful for other tutors that may face similar issues.

Acknowledgments The authors would like to thank Mr Kacper Andrzejczak for his effort during the lessons.

References

1. Industrial Robotics Market by Type (Articulated, Cartesian, SCARA, Cylindrical, Parallel), Application (Automotive, Electrical and Electronics, and Metal and Machinery), Component (Controller, Sensors, Drive), and Geography—Analysis & Forecast to 2022. http://marketsandmarkets.com, http://www.marketsandmarkets.com/Market-Reports/Industrial-Robotics-Market-643.html
2. Grand View Research: Service robotics market analysis by application (Professional, Personal) and segment forecasts To 2020. http://www.grandviewresearch.com/industry-analysis/service-robotics-industry
3. Sett, A., Vollmann, K.: Computer based robot training in a virtual environment. In: 2002 IEEE International Conference on Industrial Technology, Bangkok, Thailand (2002)

4. Yang, X., Zhao, Y., Wu, W., Wang, H.: Virtual reality based robotics learning system. In: Proceedings of the IEEE International Conference on Automation and Logistics, Qingdao, China, Sept 2008
5. Canas, J.M., Martn, L., Vega, J.: Innovating in robotics education with Gazebo simulator and JdeRobot framework. XII Congreso Universitario de Innovacion Educativa en las Ensenanzas Tecnicas (2014)
6. Lopez-Nicolas, G., Romeo, A., Guerrero, J.J.: Simulation tools for active learning in robot control and programming. In: EAEEIE Annual Conference (2009)
7. Jung, S.: Experiences in developing an experimental robotics course program for undergraduate education. IEEE Trans. Educ. **56**(1) (2013)
8. V-REP project website. http://www.coppeliarobotics.com
9. Rohmer, E., Singh, S.P.N., Freese, M.: V-REP: a versatile and scalable robot simulation framework. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on Robots and Systems (2013)
10. LabVIEW product website. http://www.ni.com/labview
11. Lua project website. http://www.lua.org
12. LabVIEW interface for V-REP. http://www.coppeliarobotics.com/contributions/labview.zip
13. V-REP's overview presentation. http://www.coppeliarobotics.com/v-repOverviewPresentation.pdf
14. Raptors (a Mars Rover project) website. http://raptors.p.lodz.pl/
15. Lerusalimschy, R., de Figueiredo, L.H., Celes Filho, W.: Lua an extensible extension language. Softw. Pract. Exp. **26**(6), 635–652 (1996)
16. BubbleRob tutorial. http://www.coppeliarobotics.com/helpFiles/en/bubbleRobTutorial.htm
17. Arduino Mega 2560 product website. https://www.arduino.cc/en/Main/ArduinoBoardMega2560
18. Petrovic, P.: Having Fun with Learning Robots. In: Proceedings of the 3rd International Conference on Robotics in Education, RiE2012. September 2012, pp. 105–112. MatfyzPress, Czech Republic (2012)
19. Swenson, J., Danahy, E.: Examining influences on the evolution of design ideas in a first-year robotics project. In: Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education, Padova (Italy), 18 Jul 2014
20. Webots project website. https://www.cyberbotics.com/overview
21. Gazebo project website. http://wiki.ros.org/gazebo

Applied Social Robotics—Building Interactive Robots with LEGO Mindstorms

Andreas Kipp and Sebastian Schneider

Abstract Teaching Social Robotics is a requiring and challenging task due to the interdisciplinary of this research field. We think that it can not be taught in a solely theoretical manner. To help students to gain more interest in the topic and to foster their curiosity we restructured a paper club like lecture to create a bridge between a theoretical topic and practical applications. This paper describes our approach to create a lecture covering theory, methods and how to transfer those to applied informatics. Described is the given theoretical input and how students learn to transfer these to a robot they build on their own. We also evaluate how the new structure was accepted and what lessons can be learned for this lecture style.

Keywords Social robotics · Educational toys · Applied techniques

1 Introduction

In a not so distant future robots might be an integral part of our daily life. The commission for innovation and research of the German government has prognosticated that the consumer and service market will be the most growing economy in the next years.[1] Hence, there is a requirement for robots that are capable of interacting with people and working in domains that were up to now staffed by humans. Thus, engineers designing and building new robots need a broad knowledge from different disciplines (i.e. sociology, psychology, computer science). This interdisciplinary demand is currently brought together in the research on social robotics. However, there is no common curricula for students that are interested in this field of research

---

[1] http://www.e-fi.de/fileadmin/Gutachten_2016/EFI_Gutachten_2016.pdf, visited 03/11/2016.

A. Kipp (✉) · S. Schneider
Bielefeld University, 33619 Bielefeld, Germany
e-mail: akipp@techfak.uni-bielefeld.de
URL: http://aiweb.techfak.uni-bielefeld.de/

S. Schneider
e-mail: sebschne@techfak.uni-bielefeld.de

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_3

yet. We see that early stage researchers coming from an engineering background have to struggle with many different theories, publications and methodologies.

In the past years we have offered courses to teach social robotics. Students read different publications connected to one of the sub areas of social robots. Those areas were i.e. robot design, emotion expression, anthropomorphism, applications and evaluation methods. The students were preparing on of the seminar sessions including a presentation of the publication, a discussion and a handout. We encountered that the seminar sessions often did not match the actual details of the topics. Thus, the association between different topics and the general scope of the seminar were difficult to grasp for the students and obstructs students to continue with this line of research. We encountered that a theoretical approach to teach social robotics is not sufficient to understand the concepts and a more hands-on approach is needed.

To overcome the difficulty of the materials and the complex access to knowledge about social robotics, we restructured the seminar in a modern fashion. We have changed the seminar to a course that covers topics on social robotics in an interactive lecture like structure. Each lecture is accompanied by an exercise where a given technique or method is directly integrated into a practical hands-on part. A group project concludes the lecture part by applying all learned elements into one practical and functional social robot. Thus, we wanted to teach how the theoretical input can be explored using a practical approach.

For the practical part the LEGO© Mindstorms EV3[2] educational set is used, allowing students to easily create robotic systems that can act and be perceived as small social entities. The robot scenario is based on the Tamagotchi developed by Bandai. This small toy uses different needs and actions to mimic a small autonomously behaving pet the user has to care about. The idea for the project was to take basic elements and behaviors from the Tamagotchi and let the students understand and transfer those to a socially perceivable LEGO robot.

In this work we want to present our concept for teaching social robotics on a university level. We will present our lecture structure, content, and methods we have used. At last, we will give a discussion about our experiences teaching social robotics, the effectiveness of our lecture paradigm, and reflect on comments and feedback from our students.

2 Related Courses and Teaching Methods

In this section we want to describe how other universities are teaching social robotics. We have searched for courses using the keywords ‘human-robot interaction’, ‘social robotics’ and ‘lecture’.

The social robots lab in Freiburg offers a seminar on social robotics.[3] Student’s learn how to conduct a literature review, read papers, and learn about state-of-the-

---

[2] http://education.lego.com/MINDSTORMS-EV3.

[3] http://srl.informatik.uni-freiburg.de/ss15seminarsocialrobotics, visited 03/10/2016.

art methods. Finally, they give a presentation of their results during a block seminar and write a summary about a paper. The content of the paper were mostly tracking, motion, and path planning.

The GeorgiaTech has offered a course on HRI.[4] The lecture covers a wide range of topics on the emergence of social intelligence and the state-of-the-art on building systems with social intelligence (e.g. Anthropomorphism, Embodiment, Experimental Design, Intentional Action, Collaboration, Teamwork, Turn taking, Dialog, Emotional Intelligence, Social Learning, Telepresence, Assistance). The lecture is accompanied by a final group project.

The Indiana University offered a course on HRI Design.[5] Topics were Classifying HRI, Evaluating HRI, Autonomy and Perception, Interfaces, Enhancing Interfaces, Robot Teams, Museum Robot, and Search and Rescue. The lecture was followed by a final project. During the course, students had to complete readings, quizzes, and labs on how to design a HRI system. Prerequisite were programming knowledge in C and JAVA. The assignments were a discussion where students have to submit half a page summary of the class paper, pros and cons, and questions. Quizzes cover the reading material using multiple choice and true-false questions. Lab assignment were conducted on a mobile robot outside of class.

At last we want to mention the lecture Principles of Human-Robot Interaction from Carnegie Mellon University[6]. Topics are: Social Robotics, Multi-modal, human-robot communication, Human-robot interaction architectures, Sensors and perception for HRI, Museum robotics, Educational robotics, Urban Search and Rescue, and Quality of Life Technologies. Students have to attend the course, read papers, answer questions related to the papers, and do a semester-long group projects.

All presented courses, except the seminar taught at the University of Freiburg, have a similar structure and similar topics. However, the description of the topics for the course are still a bit broad. This makes it hard to compare the content of i.e. ‘Autonomy and Perception’ to “Sensors and Perception for HRI”. The difficulty to distinguish the different topics of social robotics reflects the interdisciplinary of this research field. Hence, all lectures use different books or publications as the reading material for the students (there is only one publication that all of them are using as a Ref. [1]). This leads to the fact that students studying at different universities will read different references for the same subjects. We do not think that every seminar should have the same content at every university. The diversity on topics is important to give students the choice to think about to which university they apply. However, we see a demand to define a core set of topics that should be taught in a introduction course on social robotics.

Therefore, we want to report how we went from a reading-based seminar to a lecture based-hands-on course. Using this process we generated ideas how to capture the different topics of social robotics into a new curricula.

---

[4] http://www.cc.gatech.edu/~athomaz/classes/CS7633-HRI/, visited 03/10/2016.

[5] https://www.rose-hulman.edu/~berry123/Courses/HRI/HRI%20Syllabus.pdf, visited 03/10/2016.

[6] http://www.cs.cmu.edu/~illah/ri899.html, visited 03/10/2016.

3 Previous Courses

In the past years we offered several courses for students to start gathering knowledge in the field of social robotics. Most of the courses were held in a paper club style to introduce to the different subtopics of social robotics.

Throughout the lecture each student had to prepare a given topic and to present it to its fellow students. To fulfill the course the topic had to be documented in a written style and handed in at the end of the term. In the first session of the course details on the structure of the lecture as well as a basic introduction on the topic of social robots were provided. Afterwards the different subtopics got briefly introduced and distributed between the students. The common topics used for the paper club lecture can be seen in Table 1.

Each student prepared their topic themselves and presented it in the paper club. The presentation concluded with a discussion allowing fellow students to ask questions on the topics or to discuss possible conjunctions with following topics. Afterwards the students wrote a short documentation for their topic to be handed in at the end of the term. The documentation was the main element to successfully complete the course. There was no written or oral exam due to the fact, that the course was part of an individually chosen block of the students study regulations.

One big problem with this structure was the decreasing motivation to participate in later presentations, especially for those who had already presented their topic. The seminar started with around 14 students in the first sessions. Throughout the term this number decreased to about five to six students being present for the last presentation.

Another problem was the lack of fortification of knowledge and the transfer to other topics. Due to the fact that the presented information where not prompted afterwards in an exam style, many students did not tend to foster their interest.

Table 1 Topics covered in the paper club on the topic of social robotics

Topic
Anthropomorphism [2]
Forms and function of robots [3]
Uncanny valley [4]
Relations between forms and robots [5]
Perception of behaviors [6]
Attitudes towards robots [7]
Mental models for robots [8]
Socially assistive robotics [9]
Evaluation of HRI [10]
Long-term interaction with social robots [11]
Emotion models [12]

4 The Idea with a Goal: Applied Social Robotics

Based on the experiences from the previous course, we decided to change the structure for the next term and to help students in finding more meaningfulness. We wanted to foster the transfer of gathered information and knowledge into a more practical approach.

To achieve this we decided to build a new course structure around a project in which students themselves build small social robots. Because the course is open for Bachelor students that had just started their university career as well as for Master students that already had come in contact with more complex topics, we decided to use the LEGO Mindstorms platform. This technology allows, even without deeper technological understanding, to easily create small robotic systems. Nevertheless it also offers a wide range of exploration and artistic freedom for users with more expertise.

To provide a base scenario we decided to let the students create robots behaving equally to the Tamagotchi toy pet (see Fig. 1). These small devices mimic behaviors of pets by demanding attention and care (see Sect. 5.1). Hence, our teaching idea was to help students understand how the Tamagotchi can be seen as a social entity, how its behaviors can be described, and how such elements can be transferred to small robots build with LEGO.

Our overall goal was to evaluate how students deal with the new structure. We proposed that creating a conjunction between a dry theory and a practical approach

Fig. 1 The Tamagotchi device. A small child's toy capable to mimic behaviors to foster engagement. It can request attention and care due to visual and audio feedback

can help students to gain an easier access to such a topic. Additionally, such a course structure can promote creativity and result in an interesting outcome concerning the created robots.

5 Structure of the Course

For the structure of the course we decided to use three different parts: a lecture on topics of social robotics, an exercise to introduce the technological part and give some transfer ideas between topics and technology, and a project in the end of the term with focus on applying learned elements into a real robot behaving in a social manner.

The first part is a lecture. Throughout each session the lecture covers a given topic (see Table 2) that is presented by a lecturer. At the beginning of a new topic the last topic is shortly reviewed and students have to solve simple tasks like answering questions. To keep the students interested we used some interactive teaching methods that should encourage students to directly work with the knowledge. For example we use a method called *Mumble Time*. This method helps students to exchange their knowledge on the current topic with a partner and to collect ideas for a discussions with the whole group. Students that don't like to discuss in a broad group can exchange ideas with the partner and let them forward these to the group. The techniques used are based on smaller and bigger groups and work as well as working with shared contents of the presented topics. At the end of each lesson the topic is summed up and a discussion to ask questions, on possible ideas on how to apply the topic, or how the conjunction to a follow up topic can be is started. The lesson concludes with an impulse outlook for the following exercise and the upcoming topic.

The second part is the exercise. These exercises mostly cover how the technology of the LEGO Mindstorms can be used to create small robotic devices. Here the programming part gets explained as well as direct practical realizations by the students. As programming language we selected JAVA and the LEJOS[7] API. This API allows to develop control algorithms for the LEGO EV3 control brick with easy to understand elements. Throughout each exercise students work together in groups with up to three people and try to solve given tasks. These tasks cover building small moving vehicles or more complex tasks like creating emotions with the given parts.

The third and most practical part is the group project at the end of the term. Each group consists of up to four students. For every group a complete LEGO Mindstorm education set is issued to create their own robot. As project goal we defined that each robot needs to include behaviors according to those of the Tamagotchis. For the full project every group has time span of six weeks. Throughout this time frame the group has to create their own robot, program its behaviors, and to create a documentation of the project process. In the last session of the term, the groups have to present their

---

[7] http://www.lejos.org/.

Table 2 Topics to be covered in the lecture part of the seminar. Each topic is presented by a lecturer. Throughout the lecture students use group work to discover and discuss the different topics

Topic	Content and lecture	Teaching method
Introduction to the course	Information about the lecture, exercise and the group project	–
Introduction to social robotics	Outlook on research, platforms, What makes a robot social? [1, 13]	MumbleTime, Open discussions
Social agents and control structures	Introduction to agents, models for agents, Definition of control structures [14]	From mumble time to group presentations
Anthropomorphism and social actors	What means Anthropomorphism? How do humans perceive robots and agents? What means "to act socially"? [2, 15]	Group work and cross presentation
Form and design for robots	From technical to human-like robots? Design choices for robots, Examples of research platforms [4, 5]	Mumble time and collection/discussion of good, bad, ugly designs
Internal models and emotions	How to model behaviors, controlling internal states, How to express emotions? [12, 16, 17]	Hands on: How to build emotional elements for robots
Applications for social robotics	Information on ongoing research, already applied robotic systems	Group posters and presentation
Studies: How to for evaluation	Basics about evaluations, What is a research question? How to create a study? [18, 19]	Group discussion about what to evaluate based on the upcoming project

robot by providing information on their idea behind their robot, how they build and developed it, and give a prospect about possible enhancements or additions.

To fulfill the course each group has to prepare a documentation containing the information provided in the final presentation as well as a documentation of the building and programming process.

5.1 *Tamagotchi: Behaviors of a Small Toy*

To provide a scenario that defines a broad range of behaviors but leaves space for individual ideas, we decided to use the Tamagotchi toy device and its features as basis.

The Tamagotchi published by Bandai in the 1990th is a small digital toy that was created in Japan by Akihiro Yokoi. The small egg like toy has a digital display and three buttons. The toy is programmed to mimic behaviors of a small pet. With the buttons the user can manipulate and interact with the digital pet. The pet itself is based on simple wishes and desires. It needs to be *fed with different food, cleaned, disciplined, cheered*, or *cared about* in case it got ill. With the different buttons the user can initiate different actions according to the needs of the pet. The goal is to understand the different needs and to satisfy them. If not cared about correctly the pet may die and a new pet needs to be bred.

Throughout the lecture the Tamagotchi is used to compare topics of social entities with the behaviors of the toy. For this students got introduced to the Tamagotchi and were advised to understand how the digital pet works. Our goal was to teach students how to transfer the Tamagotchis behaviors into ideas for a social LEGO robot. Based on understanding the toys background the students should reverse engineer the behaviors and then transfer them to their robots. For the project and the transfer we focused on the behaviors concerning *feeding, cleaning, cheering*, and *taking care*.

Each group could themselves decide how such behaviors are expressed with the available parts of the provided set. We only advised each group that their ideas should be understandable by other persons that only interact with the robot, but who had not programmed the behaviors themselves.

5.2 The Group Project and the Results

For the project phase a total of six weeks was determined. In these weeks the student groups should think about how they could create the Tamagotchis behaviors, how to build their robot, and how to program the control mechanisms. Each team was free to explore how the given LEGO elements can be used and how their ideas could come to life. Every group was advise to document the steps taken from starting with ideas until the final robot was created. This should help to generate content for the documentation to be handed in afterwards.

Four student groups with up to three students were formed for the project phase. Each group got one full set of LEGO Mindstorms. The sets could be taken home and there was no need to bring them in or to only work in predefined slots. This allowed the groups to work freely on their own behalf. Before the final presentation all groups were randomly visited by the course supervisors to get an idea on how the groups come along. Also each group was free to ask the supervisors for help.

For the final presentation each group created a small digital presentation mentioning the ideas on how they approached the project, how they build their robot, and how they implemented the requested behaviors as well as their own. Additionally they should mention what problems occurred throughout the free working time. After all groups had presented their LEGO robot, the different robots were demon-

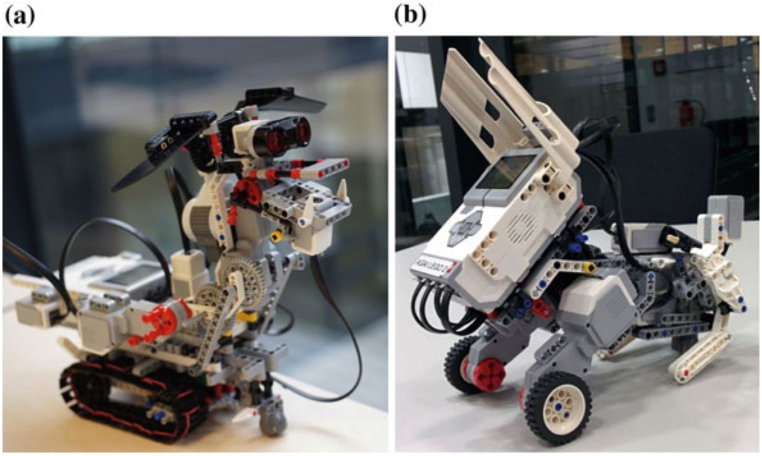


Fig. 2 Two examples of the different robots created by the project groups. Image (**a**) shows robot duck with ears to show emotions. Image (**b**) shows a dog like pet with eyes presented on the display

strated between the project groups. In this live demo the different behaviors were shown and fellow students could test the robots themselves and asks questions.

The resulting robots were quiet impressive and fully functional. Every group was able to create robots that showed the intended behaviors. Some groups focused on more complex technique robots while other focused more on simple movements with most impact on understanding the behaviors. To give an example one group created a robot that looks like a duck (see Fig. 2a). The robot is capable to move its ears up and down. This feature is used to show the emotional state of the robot. Whenever the robot is sad the ears are moving down. For the predefined states this feature is used to support the robots needs, for instance if the robot feels dirty and needs to be cleaned. This emotional state is supported by using audio files in the corresponding situation. The behavior of feeding / eating is supported by opening the mouth and by chewing whenever a given item is placed inside the mouth and therefore above the placed light sensor. In total the robot duck was programmed by using a subsumption architecture. Needs and beliefs are predefined and get selected according to values stored in the background. The designed and implemented behaviors are understandable to users and fulfill the course definitions. The second example is a dog like robot called little dog (see Fig. 2b). It can turn around, move its head, and use audio files to express its mood. One interesting feature of this robot is the cooperation needed with the human. For example if the robot gets tired, it uses audio and head rotation paired with eyes shown on the display to announce this state. The user then needs to flip the robot to the side. By using the gyro sensor this state gets detected and the robot starts sleeping and snoring. A video showing the behaviors of the little dog can be found at the CITEC video channel.[8]

---

[8] CITEC, https://www.youtube.com/citecbielefeld.

5.3 Evaluation of the Course

The Technical Faculty of Bielefeld University offers an evaluation for each course by providing questionnaires. These can be handed out to the students and will be evaluated by staff from the faculty itself. The results can then be compared between all provided courses. Unfortunately these questionnaires are optional and were not conducted for the previous lecture style. Nevertheless from colleagues we were assured that the amount of students decreases until the last sessions and that the motivation was not that high.

For our new lecture we handed out the questionnaire on the date of the presentation session. In total eleven students participated in the evaluation. We had three bachelor students, five master students, and three Ph.D students.

The results showed a positive response to the new designed course structure. One question concerns why students visit the course (see Fig. 3a). For this question multiple answers were possible. Ten students marked their interest on the topic as a reason to participate. Three students liked the idea to work in a more practical manner. Nearly all students participated in every session until the final presentation. This shows that we could keep the students interested on the topic. For the lecture part the slides used were marked as detailed and interesting (see Fig. 3b). Nevertheless an additional script would be appreciated and could help foster the transfer of gathered knowledge.

We tried to apply now teaching methods to help those students that normally would not discuss in larger groups. This commitment was positively assessed by the students group (see Fig. 4a).

For the practical part the votes showed that students liked to be creative and to transfer the topics to the LEGO robot. The results also show that the new structure fosters interest on the topic of social robotics beyond the course (see Fig. 4b).

Additionally to the questionnaire we conducted an open discussion round at the end of the presentation session. We used a method called *Five Finger Feedback*, allowing students to give one feedback concerning a scope matching a finger of the

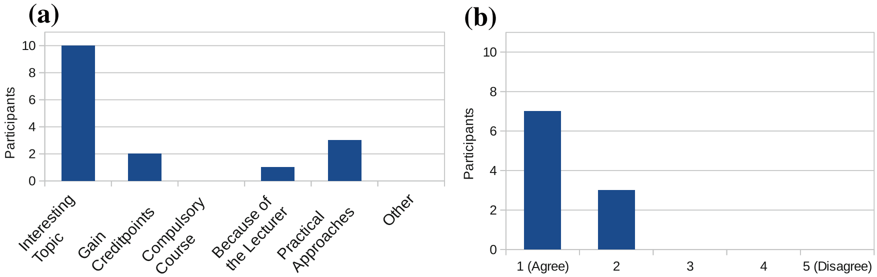


Fig. 3 Evaluation results provided by the technical faculty. **a** Why do you participate. **b** Did the provided material support the content of the lecture?

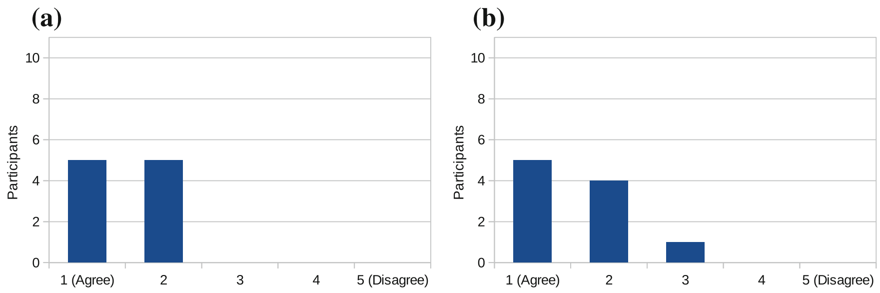


Fig. 4 Evaluation results provided by the technical faculty. **a** The lecturers are interested in teaching the topic. **b** Students are actively involved

hand: *What was good?* (thumb), *What needs to be enhanced?*, *What was not good?*, *What do you take with you?*, and *What was too short?* (little finger).

From the verbal feedback we can conclude that the new structure is very helpful to transfer theories to practical applications. It helps students to understand the associations between topics. Students reported that they liked to get a psychological point of view on computer science which was new and interesting for them. For ourselves we learned, that we need to provide more time for the exercises to foster creativity. Also we may need to hand out a script with more details on the given topics. Another point mentioned was the wish to deploy more LEGO sets to allow smaller groups and therefore to build more robots.

6 Lessons Learned

Retrospective we are happy with the new structure and we are glad we switched to the more practical part. This was our first approach to this new type and we learned much about what else could be enhanced. We will work on the parts mentioned in the evaluation and given by the students feedback to make the seminar more interesting and understandable. Also we will extend the number of LEGO sets to increase the number of further students. Additionally, we will define some criteria to rate the results and to evaluate if the provided lecture style and the given specifications match the outcome.

In general for teaching a topic that is complex and that combines many subtopics, the idea to create a bridge between theory and practical applications offers a big benefit for both, students and lecturers. With the direct transfer from lecture to exercise students can more easily adapt to gathered knowledge. Also the connection between subtopics becomes more clear. At some point for our teaching method this conjunction resulted in interesting discussions. This also helps us, the lecturers, to question the theory ourselves.

Nevertheless a practical part can help to understand theory more easily, the lecture part should also offer a good proportion of knowledge. This lecture part should also offer some impulses for the students to employ themselves even more and to create more transfer ideas for the practical parts and even other related topics.

Acknowledgments This research/work was supported by the Cluster of Excellence Cognitive Interaction Technology CITEC (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

1. Fong, T., et al.: A survey of socially interactive robots. Robot. Auton. Syst. **42**(3), 143166 (2003)
2. Epley, N., et al.: On seeing human: a three-factor theory of anthropomorphism. Psychol. Rev. **114**(4), 864 (2007)
3. DiSalvo, C.F., et al.: All robots are not created equal: the design and perception of humanoid robot heads. In: Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, pp. 321–326. ACM (2002)
4. Mori, M., et al.: The uncanny valley [from the field]. Robot. Autom. Mag. IEEE **19**(2), 98–100 (2012)
5. Powers, A., et al.: Matching robot appearance and behavior to tasks to im- prove human-robot cooperation. Human-Computer Interaction Institute, p. 105 (2003)
6. Moshkina, L., et al.: Human perspective on affective robotic behavior: a longitudinal study. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005), pp. 1444–1451. IEEE (2005)
7. Bartneck, C., et al.: The influence of peoples culture and prior experiences with aibo on their attitude towards robots. Ai Soc. **21**(1–2), 217230 (2007)
8. Lee, S.-l., et al.: Human mental models of humanoid robots. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005, pp. 2767–2772. IEEE (2005)
9. Feil-Seifer, D., et al.: Defining socially assistive robotics. In: 9th International Conference on Rehabilitation Robotics, 2005, ICORR 2005, pp. 465–468. IEEE (2005)
10. Burghart, C., et al.: Evaluation criteria for human robot interaction. In: Companions: Hard Problems and Open Challenges in Robot-Human Interaction, p. 23 (2005)
11. Leite, I., et al.: Social robots for long-term interaction: a survey. Int. J. Soc. Robot. **5**(2), 291308 (2013)
12. Picard, R.W., et al.: Affective Computing, vol. 252, ch. 2. MIT press, Cambridge (1997)
13. Dautenhahn, K.: Socially intelligent robots: dimensions of human robot interaction. Philos. Trans. R. Soc. Lond. B Biol. Sci. **362**(1480), 679704 (2007)
14. Russell, S., et al.: A modern approach. In: Artificial Intelligence, vol. 25, p. 34. Prentice- Hall, Egnlewood Cliffs (1995)
15. Reeves, B., et al.: The media equation: how people treat computers, television,? new media like real people? places. Comput. Math. Appl. **5**(33), 128 (1997)
16. Norman, D.A.: Emotional design: Why we love (or hate) everyday things. In: Basic books (2005), Chap. 6
17. Marsella, S., et al.: Computational models of emotion. In: A Blueprint for Affective Computing—A Sourcebook and Manual, pp. 2146 (2010)
18. Bartneck, C., et al.: Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. Int. J. Soc. Robot. **1**(1), 7181 (2009)
19. Feil-Seifer, D., et al.: Benchmarks for evaluating socially assistive robotics. Interact. Stud. **8**(3), 423439 (2007)

Offering Multiple Entry-Points into STEM for Young People

Wilfried Lepuschitz, Gottfried Koppensteiner and Munir Merdan

Abstract Enrollment in the STEM fields (science, technology, engineering and math) is not keeping pace with the need. Recent reports indicate a decrease in the number of graduates from STEM fields and a shortage on the job market. Considering these issues, particular attention has been paid developing innovative methods and tools for improved teaching of STEM themes. This work presents an approach involving multiple entry points for young people to engage in the STEM fields. This approach is manifested in the non-profit association Practical Robotics Institute Austria (PRIA) with its activities designed to fill STEM gaps in the Austrian education system and to bring innovative engagement that cannot be found in the classrooms. Thus, STEM literacy is fostered as well as the development of systems thinking, problem solving, and teamwork skills.

Keywords STEM · After-school program · Camps · Workshops · Non-profit association · Research · Education

1 Introduction

Science, Technology, Engineering, and Mathematics (STEM) education is a recently emerged paradigm that focuses mostly on science and mathematics disciplines, but also includes technology and engineering [1]. STEM education has evolved into a meta-discipline and thus an integrated effort that removes the traditional barriers between these subjects. Instead it focuses on innovation and the applied process of designing solutions for complex contextual problems using current tools and tech-

---

W. Lepuschitz · G. Koppensteiner · M. Merdan (✉)
Practical Robotics Institute Austria, Wexstrasse 19-23, A-1200, Vienna, Austria
e-mail: merdan@pria.at
URL: http://www.pria.at

W. Lepuschitz
e-mail: lepuschitz@pria.at

G. Koppensteiner
e-mail: koppensteiner@pria.at

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_4

nologies [2]. It shall help individuals in developing different strategies in order to solve interdisciplinary problems and gain skills and knowledge as they are engaged with STEM related activities through formal and informal learning programs [3, 4].

Currently, there is a shortage of student interest in STEM, and a lack of a well-trained teaching force in those areas [5]. Austria, like many other industrialized countries, suffers under the known phenomenon that the affinity of the youth to STEM professions decreases with the degree of achieved wealth in the country [6]. Almost 52 % of Austrians rate it "as unimportant for a personal daily routine to know something about science and research", and Austria therefore represents a front-runner in a negative sense out of 32 European states [7]. Already today in Austria, eight out of ten industrial companies have problems to find qualified personnel in the fields of engineering, production, research and development [8]. In a direct European comparison regarding the youth interest in a STEM career Austria holds one of the last places, although two thirds of university graduates, which studied in a technical field, found qualified jobs within three months after graduation [9]. Clearly, there is a need to educate large numbers of students who can fill these well-paid STEM jobs. However, engaging students in high quality STEM education requires programs to include rigorous curriculum, instruction, and assessment, to integrate technology and engineering into the science and mathematics curriculum, and also to promote scientific inquiry and the engineering design process [2].

There are a number of strategies and programs targeted to increase the number of scientists, engineers, technologists and mathematicians. Numerous initiatives from USA call for an extensive effort to reform STEM education and cultivate the next generation of skilled scientists, engineers, technicians, and science and mathematics educators [10, 11]. In Europe the EU initiative inGenious[1] aims to improve the image of the STEM fields and careers among young people and encourage them to think about the wide range of opportunities provided by working in science. It is repeatedly argued that long-term economic growth depends on a country's success at fostering young people's achievement in the STEM subjects [12].

However, increasing student competencies in science subjects is not an easy task. Several reports have stated that deductive, teacher-led lessons are still the norm [13]. This traditional approach is criticized and there are many calls for a more "active" familiarization with scientific contents [14, 15]. Research indicates that a large percentage of students' STEM learning has the potential to happen in places outside the formal classroom such as homes, after-school programs, camps, science clubs and fairs, museums, outreach labs as well as everyday experiences. Positive implications for the students' STEM learning in this context is proven as students who are engaged in informal STEM activities demonstrate higher reasoning abilities than students who do not participate in such activities [16, 17]. The report of the European Commission views the resurgence of science education as dependent on more inquiry-based teaching and, significantly, increased opportunities for the cooperation between the formal and informal sectors [18]. Consequently, emerging, innovative programs are providing engineering students with an increasingly wide variety of

---

[1] http://www.ingenious-science.eu/.

learning opportunities outside of the formal academic curricula enabling students to "experience engineering" in an authentic environment. Generally, effective out of school programs, as a one promising avenue to support students in learning and pursuing STEM careers, should be student-centered, employ cooperative learning strategies, and foster skills and attitudes towards STEM through "authentic, hands-on activities" [19].

A variety of programs alternative to traditional schooling have gained prominence in education communities [20] and a growing number of informal programs have been created to take advantage of this opportunity while addressing the need to increase participation in science and engineering [21]. Besides large, dedicated blocks of time, these informal programs have other unique benefits over formal schooling. After-school program activities have become a means for students to better understand scientific concepts and processes and allow them to acquire scientific inquiry skills and develop scientific reasoning [22, 23]. Thus, opportunities are provided for student-directed, collaborative projects where the participants learn experimentally, creating knowledge through the transformation of their own experience, and where they are immersed within the world they are learning about [24]. Furthermore, studies have shown that after-school programs can offer opportunities not available during day school and that engaging in after-school activities fosters the students' intrinsic motivation putting forth more concerted efforts in these settings [25]. After-school instructional environments offer also a greater variety of topics to explore, making STEM concepts more real to students and providing a variety of engaging and interesting activities to develop STEM skills. In this kind of environment, students learn how to collaborate and communicate with their peers and teachers in ways different from their interactions in regular classrooms [4, 26].

This work presents the non-profit association Practical Robotics Institute Austria (PRIA) as an approach for offering multiple entry points for students of different ages to engage in the STEM fields and entrepreneurship. A broad range of activities is provided for teaching and learning robotics and ICT thus supporting flexibility, experimentation, and playing with technology. The paper is structured as follows: The following section briefly introduces PRIA and its aims and organization. Section 3 describes the possibilities of PRIA offered to the students of the different school levels. Finally, the impact of the PRIA is described in the fourth section and a conclusion is given in the fifth section.

2 Practical Robotics Institute Austria

The Practical Robotics Institute Austria (PRIA) is a non-profit association with the vision to prepare and motivate next generations of researchers, engineers, and scientists. The aim to promote scientific and technical excellence in schools using robotics and ICT is reached through extensive educational and research programs. The educational activities focus on the involvement of pupils and students into complex projects and problem solving processes. This involves the development of new teaching methods, which stimulate learning and experimenting as well as realizing one's own ideas, for increasing the interest of children and adolescents in science, technology and innovation as well as the encouragement of entrepreneurial thinking. The research activities are focused on executing projects concerned with the development of innovative control architectures and technologies for robotics and industrial automation as well as product and prototype development in the fields of electronics and ICT. Generally, PRIA merges the fields of research and education integrating the achievements and findings from research in education and vice versa. PRIA uses robotics as one of the basic tools for introducing students to the STEM fields as it is an interdisciplinary domain promoting the development of systems thinking, problem solving, self-control, and teamwork skills [27].

PRIA was founded in 2012 by Dr. Gottfried Koppensteiner together with Dr. Munir Merdan and the students Christoph Krofitsch and Reinhard Grabler. Before founding PRIA, Dr. Koppensteiner already carried out several projects involving high school students in science and research during his employment at the Vienna University of Technology. Also, he organized the European Conference on Educational Robotics (ECER), a scientific conference for high school students involving also a robotics competition, for the first time in 2012. To intensify the involvement of students into research and to have a legal body for acquiring the necessary budgetary means, PRIA was founded. While both Dr. Koppensteiner and Dr. Merdan could look back already on a successful scientific career, the other two founding members Mr. Krofitsch and Mr. Grabler just had graduated from high school. Nevertheless, due to their former engagement in scientific projects during their school time, they were eager to support the creation of PRIA and remained as employees ever since. PRIA is located at their former school called Technologisches Gewerbemuseum (TGM), which is Vienna's largest technical high school. Around 2400 high school students aged generally between 14 and 19 years old attend this school. Additionally 400 students attend the evening school for adults.

3 Young People Engaging in STEM

PRIA offers multiple entry points for young people to become acquainted with STEM and entrepreneurship. PRIA provides a program with activities for students from primary, middle and high school as well as for those attending university (see Fig. 1).

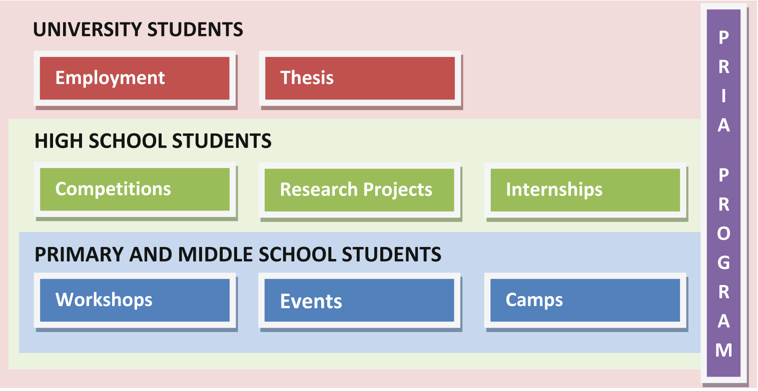


Fig. 1 Entry points offered by PRIA for young people of different school levels

3.1 Entry Points for Primary and Middle School Students

3.1.1 Workshops

Primary and middle school students can engage themselves in robotics workshops. At the very beginning the students built up their first simple robot in electronic lessons to get in touch with an easy level of robotics. This robot consists of a body and batteries along with a simple electronic circuit with two motors and two light transistors. The robot is an intelligent light follower, strongly derived from Braitenberg's simple robot [28] and the BYO-Bot.[2] In this workshop the students learn about electrical and mechanical aspects. In further workshops the students are exposed to the technologies used in LEGO Mindstorms or Botball. The educational repertoire includes graphical programming but also easy programming guides for C-language, teaching them about vision systems as well as sensors, motors, and controllers. Depending on the project and the demands of a school, workshops are offered in various lengths. A university graduated professional of PRIA serves as primary contact for the courses, facilitating activities and managing content and organization issues. Extraordinary undergraduate students employed by PRIA provide instructional support. They serve as mentors and role models for the participants in the workshops. During the school year, PRIA offers workshops for schools in the frame of publicly funded projects (such as ER4STEM[3]).

---

[2] https://bbstore.mycafecommerce.com/product/hand-assemble-byo-bot.

[3] http://www.er4stem.com.

3.1.2 Summer and Winter Camps

Apart from the publicly funded workshops, so-called summer and winter camps are offered in the official school holidays. This program integrates successful ideas from similar other programs [29]. Mobile robots are used as a motivating and interesting tool for studying machines design and construction, software, as well as communications systems. The program allows participants to connect theory with practice and exercise team work, project management, problem solving, and communication skills in a stimulating setting. The students are mentored in different ways of performing their tasks and how to solve the problems from stage to stage. Throughout these activities, the student groups develop different ideas and concepts in order to accomplish their goals, issues, and problems. Currently the winter camps are offered during one week in February and summer camps are offered during one or two weeks at the end of August. Furthermore, PRIA is regularly carrying out workshops in various holiday camp activities organized by other institutes or companies.

3.1.3 Events

PRIA regularly organizes events for bringing science to the school students. In the frame of the regionally funded project STEMoFuture, PRIA organized two times the "Long Night of Technology" at the TGM. Students from primary and middle schools were informed and encouraged to visit these events as they incorporated exhibitors from several institutes such as the Vienna University of Technology or the Austrian Institute of Technology. Furthermore, the TGM itself presented some of its departments to the possible future high school students. Moreover, primary and middle schools are invited every year to attend the ECER as guests. By seeing older school students engaging themselves in robotics, the younger ones realize that technology is not only for adults but can be pursued also at a younger age at an advanced level. Likewise to the primary and middle school students, of course also older students are invited to attend events such as a "Long Night of Technology".

3.2 Entry Points for High School Students

3.2.1 Competitions

Each year PRIA organizes the European Conference on Educational Robotics (ECER[4]). It is a well established international scientific conference for high school students having the official European Regional Botball Competition as core activity. Botball[5] is a team-oriented robotics competition, at which each team receives a

---

[4] https://pria.at/en/ecer/.

[5] http://www.botball.org/.

standardized robotics set consisting of metal and Lego parts, sensors and actuators as well as two robot controllers. New competition tasks are developed every year and published at the Botball workshop at the beginning of a new season, giving every team the same time for developing their strategies and robots. In the preparation for ECER, PRIA offers annually a large Botball workshop free to attend for ECER participants. Technical details of Botball are discussed and the game rules for the competition at ECER are explained. Furthermore, the workshop also contains a session concerning scientific writing. Generally, students develop their robots to solve the tasks of the game table having about three months to find a good strategy and to build up and program their robots for this strategy. At the conference the students can present their findings in engaging talks, show their robots live, and take part in the exciting robot competitions. A special focus is given to planning, documentation, and the quality of the technical solutions. Apart from the competition, special awards are granted for outstanding achievements in programming, mechanical engineering as well as documentation.

3.2.2 Research Projects and Pre-scientific Works

PRIA integrates high school students into the ongoing research projects carried out by actual researchers. Typically these projects are related to the topics of Industry 4.0, which fits well to the abilities of high school students from technical high schools such as the TGM. In such schools, the students have to compose a pre-scientific work denoted as diploma thesis. By knowing that they contribute to an actual research project, the students are assured that their work is meaningful. Real projects with meaningful outcomes have been shown to engage students, especially when real scientists are involved [30]. In case the students would like to carry out a project of their own, PRIA considers if it fits into the association's goals and gives support even if it is not directly part of a research project. Projects and pre-scientific works represent first steps for the high school students to become entrepreneurs as they have to manage their work themselves (of course with assistance of PRIA staff). This kind of participation is of advantage for all involved stakeholders of the project. The evaluation of a project in which scientists partnered with teachers found considerable benefits for teachers and scientists as well as the students; benefits that included increased knowledge and considerable enjoyment for all persons concerned [31]. Moreover, many students dont envision themselves as scientists, in part because they dont see scientists as "real people." Formal and informal opportunities to connect with scientists can help students recognize that those are "regular people", who have hobbies, families, and out-side interests [32].

3.2.3 Internships

During the summer holidays PRIA offers internships, which can be on the one hand in the frame of research projects offering an insight into ongoing research activities.

On the other hand the internship can be in the frame of the robotics workshops in PRIA's summer camps as an assistant supervisor supporting the attending young people in their endeavors with robotics.

3.3 Entry Points for University Students

3.3.1 Employment

After graduation from high school, talented young people can become employees of PRIA. Most of the current university students employed at PRIA already were involved in previous PRIA activities during their school time. The possible tasks are manifold and the work requires technical as well as social skills. Students can engage in the research as well as in the educational activities of PRIA (e.g. supervising at workshops or at summer and winter camps). Moreover, when a group of high school students carry out their diploma thesis project, university students employed at PRIA often act as their primary contact person (even though supervised by one of the PRIA researchers with university degree). Thus, they improve both their technical and their social skills. Furthermore, they can take part as co-authors at the composition of scientific papers.

3.3.2 Masters Thesis, Bachelors Thesis, Term Paper

A further opportunity for university students is to write a thesis or a term paper. Either it is a topic related to work conducted at PRIA or the student can suggest a topic. In either case an employee of PRIA with a university degree can act as an external supervisor.

4 Young People Engaging in Research

As mentioned earlier, high school students as well as university students have the opportunity to participate in ongoing research projects at PRIA. One of these projects entitled "Smart Phone Control of Robots for Education & !ndustry" (SCORE!) is concerned with the use of mobile devices for controlling robots in an easy and affordable way.

Modularity and flexibility in robot-based training and education while keeping the work-setting simple and intuitive is a key factor for involving teachers and students. Mobile devices such as smartphones offers great potential as most people are familiar with them. The smartphone has remarkable power in computation, involves convenient operations, such as camera monitoring or wireless internet, and encompasses various components that are useful for controlling robots. Part of the project

Fig. 2 Robotics controller Hedgehog and according smartphone app for programming and controlling robots

SCORE! was the development of a robotics controller denoted as Hedgehog consisting of hard- and software, which offers flexibility and extension possibilities for in the domain of robotics education. The controller works with a two-layer-architecture based on reflective (low-level control) and planning (high-level-control) paradigms, which, besides supporting educational features, makes it usable for applications in robotics research as well. An app for mobile devices is used as interface to the user, which enables robot testing by accessing sensors, motors and servos, as well as developing programs that can be downloaded to the controller to get executed (Fig. 2).

During the project, 25 high school students participated extensively in the development of the Hedgehog controller. A total of five high school diploma theses and four yearly projects were finished. Additionally, several high school students participated in the tests of the Hedgehog controller. Furthermore, three university students wrote their bachelor and diploma theses in the frame of this project [33–35] and three university students as well as one high school student, all of them employed by PRIA, took part in the publication of research papers [36–38].

A further project, entitled "Batch Process Automation with an Ontology-driven Multi-Agent System" (BatMAS), is concerned with the application of knowledge-based systems in the domain of Industry 4.0. In the frame of this project, a group of four high school students worked on their high school diploma thesis and one university student employed by PRIA contributed to a scientific publication [39].

5 Impact

PRIA's impact is significantly rising from year to year. In 2015 more than 1050 children and adolescents were reached by the activities carried out by PRIA. 35 % of them were girls and 33 % had migration background. Due to the funded projects, 80 % were able to participate without costs. Additionally, nearly 400 parents were also reached as they represent an important part in the decisions of the children and adolescents regarding education and career.

6 Conclusion

This paper presented the entry points provided by PRIA for young people to the STEM fields. In principle the fields of research and education are merged offering multiple possibilities for young people to become acquainted with STEM encompassing workshops, camps, events, followed by competitions, research projects, and finalized with thesis, mentoring and employment. This provides opportunities for students on the one hand to increase their STEM literacy and on the other hand to acquire both problem-solving skills and experience relevant for their future development. The PRIA approach highlights how innovative and effective ongoing activities can improve the opportunities for students and help fix the broken pipeline in STEM education.

Acknowledgments The authors acknowledge the financial support from the European Unions Horizon 2020 research and innovation program under grant agreement No. 665972.

References

1. Bybee, R.W.: What is STEM education. Science **329**, 996 (2010b). doi:10.1126/science.1194998
2. Kennedy, T.J., Odell, M.R.L.: Engaging students In STEM education. Sci. Educ. Int. **25**(3), 246–258 (2014)
3. Lacey, T.A., Wright, B.: Occupational employment projections to 2018. Mon. Labor Rev. 82–109 (2009)
4. Sahin, A., Ayar, M.C., Adiguzel, T.: STEM-related after-school program activities and associated outcomes on student learning. Educ. Sci. Theor. Pract. **14**(1), 13–26 (2014)
5. Mataric, M.J., Koenig, N.P., Feil-Seifer, D.: Materials for enabling hands-on robotics and stem education. In: Proceedings of the AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education, Stanford, CA (2007)
6. Sjberg, S., Schreiner, C.: The next generation of citizens: attitudes to science among youngsters. In: Bauer, M., Allum, N., Shukla, R. (eds.) The Culture of Science How does the Public Relate to Science Across the Globe? Routledge, New York (2010)
7. EUK, Science and Technology Report, Eurobarometer (2010)
8. Industriellenvereinigung: ZAHLEN, DATEN & FAKTEN Arbeitsmarkt und Karrierechancen in Mathematik, Informatik, Naturwissenschaften und Technik (2013). http://www.iv-net.at/iv-all/publikationen/file_610.pdf, Accessed 19 Jan 2016
9. Wanek-Zajic, B.: Wohin nach der Ausbildung? Bildungsbezogenes Erwerbskarrierenmonitoring (bibEr), AMS info. No. **222** (2012)
10. Rising Above the Gathering Storm: Energizing and employing America for a brighter economic future. National Academy Press, Washington, DC (2007)
11. Obama Educate to innovate. https://www.whitehouse.gov/the-press-office/president-obama-launches-educate-innovate-campaign-excellence-science-technology-enon, 23 Nov 2009

12. Sawyer, R.K.: Optimising learning implications of learning sciences research. In: OECD (Ed.) Innovating to learn, learning to innovate, pp. 45–65. OECD Publishing (2008)
13. Andres, S.H., Steffen, K., Ben, J.: TALIS 2008 technical report. OECD, Teaching and Learning International Survey, Paris (2010)
14. Itzek-Greulich, H., Flunger, B., Vollmer, C., Nagengast, B., Rehm, M.: Trautwein U: effects of a science center outreach lab on school students' achievement are student lab visits needed when they teach what students can learn at school? Learn. Inst. **38**, 43–52 (2015)
15. Schroeder, C.M., Scott, T.P., Tolson, H., Huang, T.-Y., Lee, Y.-H.: A metaanalysis of national research: effects of teaching strategies on student achievement in science in the United States. J. Res. Sci. Teach. **44**, 1436–1460 (2007)
16. Gerber, B., Edmund, M., Cavallo, A.ML.: Development of an informal learning opportunities assay. Int. J. Sci. Educ. **23**(6), 569–583 (2001)
17. Nite, S., Margaret, M., Capraro, R., Morgan, J., Peterson, C.: Science, technology, engineering and mathematics (stem) education: a longitudinal examination of secondary school intervention. IEEE Front. Educ. Conf. (FIE) (2014)
18. Commission, European: Science Education Now: A Renewed Pedagogy for the Future of Europe. European Commission, DirectorateGeneral for Research, Brussels, Belgium (2007)
19. Hussar, K., Schwartz, S., Boiselle, E., Noam, G.G.: Toward a systematic evidence-base for science in out-of-school time: The role of assessment (2008). Accessed 11 Sept 2008
20. Bucknavage, L.B., Worrell, F.C.: A study of academically talented students in extracurricular activities. J. Second. Gift. Educ. **6**(2/3), 74–86 (2005)
21. Cardella, M.E., Wolsky, M., Paulsen, C.A., Jones, T.R.: Informal pathways to engineering: preliminary findings. Paper presented at 2014 ASEE Annual Conference, Indianapolis, Indiana (2014). https://peer.asee.org/20638
22. Fisanick, L.M.: A descriptive study of the middle school science teacher behavior for required student participation in science fair competitions. Published Doctoral Dissertation, Indiana University of Pennsylvania, Pennsylvania) (2010)
23. McGee-Brown, M., Martin, C., Monsaas, J., Stombler, M.: What scientists do: Science Olympiad enhancing science inquiry through student collaboration, problem solving, and creativity. Paper presented at the annual National Science Teachers Association meeting, Philadelphia, PA (2003)
24. Kolb, David A., Learning, Experiential: Experience as the Source of Learning and Development. Prentice Hall, Englewood Cliffs, NJ (1984)
25. Vandell, D.L., Shernoff, D., Piece, K., Bold, D., Dadisman, K., Brown, B.: Activities, engagement, and emotion in after-school programs (and elsewhere). New Dir. Youth Dev. **105**, 121–129 (2005)
26. Mahoney, J.L., Cairns, B.D., Farmer, T.W.: Promoting interpersonal competence and educational success through extracurricular activity participation. J. Educ. Psychol. **95**(2), 409–418 (2003)
27. Johnson, J.: Children, Robotics and Education. In: Artificial Life and Robotics, 16–21. Springer (2003)
28. Braitenberg, V.: Vehicles: Experiments in Synthetic Psychology. MIT Press (1986)
29. Stapleton, W., Asiabanpour, B., Stern, H., Gourgey, H.: A Novel Engineering Outreach to High School Education, Imagining and Engineering Future CSET Education, 2009 Frontiers in Education (FIE) Conference. San Antonio, TX (2009)
30. Stocklmayer, S.M., Rennie, L.J., Gilbert, J.K.: The roles of the formal and informal sectors in the provision of effective science education. Stud. Sci. Educ. **46**(1), 1–44 (2010)
31. Rennie, L.J., Howitt, C.: Science has changed my life!: Evaluation of the scientists in schools project (A report prepared for CSIRO.). Department of Education, Employment and Workplace Relations, Canberra, Australia. Accessed 25 Nov 2009
32. Cohen, C., Patterson, D.G.: The emerging role of science teachers in facilitating STEM career awareness (2012). Retrieved from http://nwabr.org/sites/default/files/pagefiles/teaching-STEM-career-awarenessPRINT.pdf

33. Novotny, L.: Mensch-Maschine Schnittstelle im Bereich der mobilen Robotik; supervisors: M. G. Koppensteiner, PRIA, Tesar, FH Technikum Wien (2013)
34. Krofitsch, C.: Remote Debugging Environment for Educational Robotics; supervisor: P. Vienna University of Technology, Puschner (2015)
35. Chua, D.: Entwicklung und Umsetzung von Augmented Reality im Bereich der Industrierobotik; supervisors: C. W. Lepuschitz, PRIA, Engelhardt-Nowitzki, Technikum Wien (2015)
36. Krofitsch, C., Hinger, C., Merdan, M., Koppensteiner, G.: Smartphone driven control of robots for education and research. In: IEEE 2013 International Conference on Robotics, Biomimetics, Intelligent Computational Systems, Yogyakarta, Indonesien (2013)
37. Krofitsch, C., Lepuschitz, W., Klein, M., Koppensteiner, G.: Flexible development environment for educational robotics. In: International Conference on Control, Automation and Robotics, Singapore (2015)
38. Koza, C., Krofitsch, C., Lepuschitz, W., Koppensteiner, G.: Hedgehog: a smart phone driven controller for educational roboticspaper. In: 6th International Conference on Robotics in Education, Yverdon-les-Bains, Switzerland (2015)
39. Lepuschitz, W., Lobato-Jimenez, A., Axinia, E., Merdan, M.: A survey on standards and ontologies for process automation. In: Proceedings of the 7th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, pp. 22–32 (2015)

Part II
Educational Robotics Curricula

How to Teach with LEGO WeDo at Primary School

Karolína Mayerové and Michaela Veselovská

Abstract In this paper we present a set of activities with robotics kit Lego WeDo. Activities are designed for ordinary pupils of the third and fourth grade of primary school, and not only for robotic fans. We briefly describe every task of each activity. These eight activities are iteratively confirmed sequence which we recommend to use for familiarization with the elements of the kit and also software. Set of activities and recommendations to them is the result of several iterative verifications and editing, based on observations of teaching according to our materials during informatics lessons. These activities follow the requirements of our national education program and develop several 21st century skills for successful life.

Keywords Activities · LEGO WeDo · Primary educational · Robotics · Introduction to programming

1 Introduction

In this article we present our final curriculum for educational robotics, which were inspired by many ideas. For example we followed ideas of constructionism [1, 2], managing the robotics classroom [3] or development of educational goals in psychomotor, affective and cognitive domain [4]. This pedagogical intervention is the output of our iterative research developments in the field of educational robotics. In this article we present the eight activities which we designed for pupils of primary school. We created this curriculum for the purpose of implementing educational robotics into compulsory subject Informatics in Slovakia. Therefore we based on the experiences we have gained during the realization of our dissertation project [5], and also with respect to the existing national educational curriculum for informatics

---

K. Mayerové (✉) · M. Veselovská
Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
e-mail: mayerova@fmph.uniba.sk

M. Veselovská
e-mail: veselovska@fmph.uniba.sk

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_5

[6]. So that they are now usable in average class at primary school and allow development of 21st century skills in appropriate and playfully way [7]. For our research goals we were searching for robotic kit, with which pupils can develop skills to construct robotic models and also skills to program it. Therefore we chose robotic kit LEGO WeDo. There are many another robotic activities, for example [1, 3, 8] but they are not usable for primary school subject Informatics in our country.

2 Research Methodology

One of the aims of our dissertation research was to iteratively design, implement and verify curriculum, which can introduce one of possible ways of implementation of educational robotics into informatics in primary school. In our research we specified two research questions. In this article we deal with one question: What are the aims, form and content of curriculum for educational robotics in context of primary school subject Informatics in Slovakia? So, in our dissertation research we focused on the creation, implementation and iterative verification of our curriculum, which consists from activities for primary school pupils and methodological materials for teachers. These activities are destined for informatics lessons using the LEGO WeDo construction set with its integrated programming environment for 8 to 12 years old pupils. In our research we used qualitative methods of data collection and data analysis [9], including observation (field notes, transcriptions and drawings), focus groups, audio-visual materials (pictures, photographs and recorded videos) of pupil's products. As a research strategy, we used design based research [10], which allows researchers to enter and interfere the process and thus create an enhanced intervention. We verified our activities with more than one hundred lessons in five stages during few years. We conducted our research in three different primary schools in Slovakia. In first stage of our research one researcher taught designed activities and other researcher was collecting data. In the other stages of research our activities were taught by three different teachers in three different schools and both researchers were collecting data. We used different methods for data analysis, for example open coding and constant comparison. For validation of achieved learning goals we used analysis of recorded videos, which included pupils work and reactions during realization of our activities. This way we could compare their skills and knowledge in the beginning and in the end of our intervention.

3 Our Eight Activities

For each of the eight activities we have developed methodical material for teachers. For some activities we also designed and verified worksheets for pupils. We recommended that it is beneficial to implement the first four activities in third grade and another four activities in fourth grade. The full version of our curriculum in final form, along with worksheets for pupils can be found in our doctoral thesis[1] [11]. Each methodological material contains the following seven sections: Preconditions, Aids, Goals, Competencies,[2] Time, Assignments and The recommendations. But for lack of space we mention only some of them. In our activities pupils work in small groups (two or three members). Each activity also contains the correct solutions for programming problems from worksheets, as well as examples of expected final models. In the next part of article there is detailed description of the eight activities.

3.1 Hello Robot—Activity 1

The learning object of first activity is to organise pupils' previous knowledge about robots, using discussion. These activities were described in our earlier papers [14]. At the end of the lesson, pupils should have learned and know that robots' main purpose is to help people. Robots need some energy source to move (battery, adapter and so on) and they consist of various components; robots cannot think on their own, only if we "tell them to", that is if we program them. Another goal of the lesson is to introduce pupils to working with the Lego WeDo construction kit, and thus help develop their fine motoric capacities and communication skills.

Description of Activity: In the beginning of activity we conduct discussion with pupils. This way we can easy identify their experiences, representations and concepts which they usually connect with theme robot. In the end of discussion we explain next task in which they should create their own robotic model. In the end of whole activity they introduce their robots with short presentation to other classmates. They use robotic kit LEGO WeDo and program motor to move with software in computer. In the end of activity every pair present their robot, so they state its mane and explain its functions. The rest of pupils come close to presenting pair and they listen carefully. They can also ask some questions about presented robot.

3.2 Building and Programming Small Airplane—Activity 2

This lesson's learning object is to make pupils familiar with the Lego WeDo software environment and its basic icons. It is their first encounter with commands such as *shape wait for, motor on for, motor power* or possibly with the *shape repeat*[3] command. The aim is, using attractive approaches, to allow pupils develop in them the habit of following instructions, to observe the difference between commands *shape wait for* and *shape motor on for* and to be able to apply commands in the correct

---

[1] so far only in Slovak version.

[2] are derived mainly from the skills for the 21st century skills [12, 13].

[3] The *shape repeat* command appears in the end of activities only and not every pupil necessarily manages to go through all of the activities.

order. Another objective is to improve the pupils' fine motoric skills by constructing a model and also develop social behaviour, communication and cooperation skills using group work exercises.

Description of Activity: Pupils build airplane according to instructions.[4] Then they program it several different ways. For example turn motor on (to spin one way), stop motor and turn it on to spin other way, simulate starting of motor of airplane, and so on.

3.3 What Will Transport Look Like in the Future?—Activity 3

This lesson aims at strengthening the knowledge of motor control. Furthermore, it focuses on the development of creativity and skill to put one's own ideas in practice using creative activity. Through fulfilling the assigned tasks, children look for suitable methods to find solutions, later on for effective ways for improving their models.

Description of Activity: Pupils should design, build and program model of vehicle, which people will be used in 200 years. About 15 min before the end of activity we interrupt pupils work and open presentations. This way every group of pupils can introduce their model. We designed and confirmed two types of assessment for this activity.

3.4 Rafting—Activity 4

The learning object of this activity is to develop, through playful forms, the capacity to express oneself intelligibly and realise the importance of the sequence of order, i.e. if we change the order of steps while building a particular model what consequences it may have. Pupils should comprehend the importance of the details described, in particular in the case of varied shapes and colours.

Description of Activity: In the beginning we talk about times when people were rafting on the river differently as today, etc. We can ask pupil if they know what is raft and if someone want to explain it to others. Then we explain task to pupils: they should build model of raft but in different way. We prepared model of raft, which consists from several smaller models, see on the left side Fig. 1. This model is hidden from pupils. Pupils create several groups with 3 or 4 members. Each group choose one member as observer. Only observer can see hidden model (which is situated in marked place) and he can explain its structure to other members of group. Observer can only describe mentioned model verbally, but he/she cannot show components of

---

[4] Available at Lego Education page: https://education.lego.com/en-us/lesi/support/product-support/wedo/wedo-base-set-9580/building-instructions.

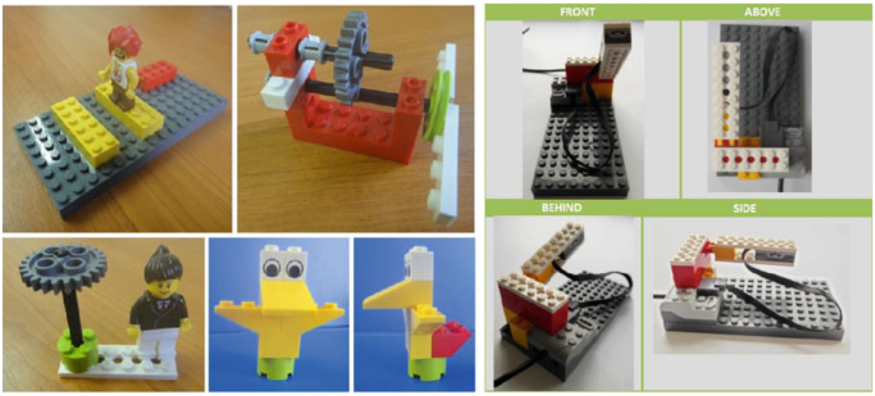


Fig. 1 *Left side*—Models from Activity 4. *Right side*—Example of the photographs of models from Activity 7

model with any part of his/her body. We have prepared four models, so each member of group can be observer. Other members of group should build hidden model based on observers' verbal explanation and instructions. If they do not understand, they can ask questions to observer. When pupils build four smaller models, they can merge them into one larger model.

3.5 Ventilator—Activity 5

The learning objective is to develop ability to determine specific elements based on their identification in the final product and following the instructions. It provides training of the capability to transfer information from a graphical representation to reality, introduces pupils to the tilt sensor functionality and related software icons. The activity involves revision of basic commands by means of gradual sequence of programming tasks and further works with the *shape wait for* command in combination with *shape numerical parameter* and later links it with *shape sensor parameter*.

Description of Activity: At the beginning of the lesson, we ask the pupils to say what they remember from previous robotics lessons. We should focus on two areas: design and programming. Pupils should say something interesting and different from what was said by previous classmates. Thereafter pupils will build ventilator according to the instructions. When pupils finished they work at model, they start to program it. At the end of the worksheet are instructions, how to move ventilator. We can solve these tasks together with all pupils, or let pupils work individually. Each group of pupils then can progress at their own pace.

3.6 Remembering the Coding the Ventilator—Activity 6

The learning objective is to strengthen knowledge learned during the previous lesson and clarification of commands provided these have not been understood properly, such as parameter values when setting *shape the motor power*, unproductive command duplicity, *shape the wait for* command in combination with *shape sounds* or *shape tilt sensor* and *shape parameter* used in *shape repeat*.

Description of Activity: The tasks in the worksheet are efficiently ordered to lead pupils to reflect on the meaning of each icon, or to reflect effectiveness different combinations of icons. So that the pupils can test programs in a worksheet, first they must build a simple model of the ventilator. When pupils solved the problem in a worksheet, they start to build their stable base for ventilator, i.e. tower which has on top the ventilator. It is possible that this activity can take two lessons.

3.7 The Administrator of Windmill—Activity 7

The learning object of this activity is to create a model based on information displayed on a photograph. Based on such resultant model, pupils are tasked to identify and track back the procedure and repeat it. Their task is to build an other model. They should express in words the functionality of the already created program, create a program based on written instructions(given in words not using icons), and also modify the already created program. Pupils test and characterise the functionality of the presented programs using tilt sensor, and finally gain a real experience with running three programs in parallel.

Description of Activity: Tasks in the worksheet are divided into three parts. In the first part pupils have to build a latch from parts at the photo. There are photographs of the finished model of the latch on the right side Fig. 1. Thereafter pupils have to build their own windmill based on their imagination. In the second part they have to work with the program. In the third part, which probably do not succeed all the pupils have to link the various programming structure in the left column with the correct description in the right column.

3.8 What Did We Learn?—Activity 8

The goal of this activity is a final verification of programming skills in the Lego WeDo environment and strengthening of the knowledge gained during previous lessons. Moreover, it focuses on the explanation of programming concepts, which have not been properly understood so far, making space for development of creativity combined with design and construction of own models based on the given criteria and simple demonstration of parallelism.

Description of Activity: At the beginning of the lesson teacher distributes the worksheet with two parts of tasks to each pupil. In the first part pupils have to write how works the program composed of icons. This second part looks same like the third part in Activity 7. Follows free creative activity. Pupils should create their own model to be controlled according to the several criteria described in worksheet. Pupils need to be lead do not spend the whole time only by building a model, but focus mainly on the control (programming).

4 Conclusion

In conclusion, we would like to summarize a few recommendations, or research findings we have gained during verification of our activities in practice. The aim of our study was to develop activities that would be applicable in ordinary primary school classes, and not only for gifted children, or for children with increased interest in technology. Although we believe that the activities with robotic kits have their place especially among these children. As we mentioned at the beginning, all the activities are designed for approximately one lesson (45 min). Of course it depends on the potential of students and teachers abilities effectively lead the lesson. This means constantly be in contact with students and walk from one group to another, constantly develop the pupils' productive compulsion to work and make effectively. From our experiences we know that activities with LEGO constitute a significant motivation for students and can enjoy working with them for long periods. In addition, the scientific literature confirms that the implementation of educational robotics in the primary education with one of the objectives being the development of playful introduction to programming basics has a great educational potential, but has only a limited use in the real schooling so far. We believe that we have created a good curriculum for educational robotics for primary school education, which can be used by teachers to develop various skills and competences in pupils, that is also an interesting and attractive introduction in educational programming.

References

1. Papert, S.: The Eight Big Ideas of the Constructionist Learning Laboratory. South Portland, Maine (1999). Unpublished internal document
2. Kafai, Y., Resnick, M.: Constructionism in Practice. Lawrence Erlbaum Asociates, Mahwah, New Jersey (1996)
3. Gura, M.: Getting Started with LEGO Robotics: A Guide for K-12 Educators. International Society for Technology in Education, United States of America (2011)
4. Pasch, M.: Teaching as decision making: instructional practices for the successful teacher. Addison Wesley Publishing Company (1991)
5. Mayerová, K.: Pilot activities: LEGO WeDo at primary school. In: 3rd International Workshop on Teaching Robotics, pp. 32–39 (2012)

6. NPI.: New national curriculum for primary school: Informatics. National Pedagogical Institute, Bratislava. http://www.statpedu.sk/sites/default/files/dokumenty/inovovany-statny-vzdelavaci-program/informatika_pv_2014.pdf, Accessed: 03 Dec 2016
7. Mayerová, K., Veselovská, M.: Robot kits and key competences in primary school. In: Information and Communication Technology in Education. University of Ostrava, Ostrava, Pedagogical Faculty, pp. 175–183 (2012)
8. Ilieva, V.: ROBOTICS in the Primary School—how to do it? In: International Conference on Simulation, Modeling and Programming for Autonomous Robots, SIMPAR, Darmstadt, pp. 596–605 (2010)
9. Creswell, John W.: Educational Research: Planning, Conducting, and Evaluating Quantitative. Upper Saddle River, New Jersey (2002)
10. Kalaš, I.: Pedagogický výskum v informatike a informatizácii (2. časť). DidInfo 2009, pp. 15–24 (2009)
11. Mayerová, K.: Miesto edukačnej robotiky v informatickej výchove (Educational robotics in primary education). FMFI UK in Bratislava, Bratislava (2015)
12. Trilling, B., Fadel. Ch.: 21st century skills: Learning for life in our times. John Wiley & Sons(2009)
13. ATCS.: Assessment and Teaching of 21st Century Skills: Defining 21st century skills. http://atc21s.org/wp-content/uploads/2011/11/1-Defining-21st-Century-Skills.pdf, Accessed 20 Jan 2015
14. Mayerová, K., Veselovská, M.: How We Did introductory lessons about robot. In: Teaching Robotics, Teaching with Robotics, Padova, pp. 127–134 (2014)

Using Modern Software and the ICE Approach When Teaching University Students Modelling in Robotics

Sven Rönnbäck

Abstract This paper presents a robotic course that was revised with the ICE (Ideas-, Connections-, Extensions levels) approach of teaching in mind. It includes practical labs where the students have to derive and implement mathematical models and verify those on a manipulator arm mounted on a small mobile robot called KUKA youBot. To support hands-on experience Simulink blocks have been developed to enable sensor readings and control of the youBot. The students implement their models using Simulink and run them to control the youBot manipulator. The labs have been used in several course settings with some modifications. The course also includes a small project where the students use and go beyond the results achieved in the labs, it relates to the *Extensions level* in ICE. Even though the approach is under development it has been tested and successfully used in a course.

1 Introduction

It is important to align teaching activities with learning outcomes, so called *Constructive Alignment* [1], and that the learner get relevant learning activities to the topic taught.

Our belief is that students can achieve deeper learning in robotics, such as forward kinematics, inverse kinematics, and dynamics, if they get hands-on experience and can feel the robot. In our robotics course we want students to achieve *knowledge*, develop *skills*, and gain *understanding* [2].

The ICE approach [3] came in mind when reconstructing the curriculum for the course *Modeling in Robotics*, 7.5 ECTS (2015), previously a bigger course named *Control Methods for Robot Applications*, 15.0 ECTS (2014). ICE presents three levels; *Ideas-, Connections-*, and *Extensions level*. The different ICE levels do not necessary need to be assigned to different grades but to different parts of a course. At *Ideas level* the students learn the theory, vocabulary, and information they can recall. At *Connections level* learning happens when students are able to relate their

---

S. Rönnbäck (✉)
Department of Applied Physics and Electronics, Umeå University, 901 87 Umeå, Sweden
e-mail: sven.ronnback@umu.se

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_6

learning to what they already know [3]—build connections between the bits and pieces, and at *Extensions level* learning happens when students take what they already know and create something new. The three different levels are often associated to different course grades, where *Ideas level* is related to pass grade, and *Extensions level* to the highest course grade.

We map *Ideas level* to lectures, recitations, and exercises. In the laboratory work the students start to put the bits and pieces together, and we consider this to be related to *Connections level* since it is where students actually through implementation and testing gain experience on key concepts taught in lectures and exercises. The labs are assignments, *understanding performances* [4], that are assessed through written reports.

The project part, which is the last part in the course, students choose one project among proposed ones and applies their understanding in a new setting, *understanding performance* [4], to stretch knowledge, skills, and understanding.

The project part can be seen as the *Extensions level* according to ICE; the student apply and create something new based on the concepts from *Ideas* and *Connections levels*.

2 The Course and Its Hardware

Labs and projects are centered around the manipulator arm on the KUKA youBot [5]. YouBot is a small mobile platform equipped with a lightweight five degrees of freedom robotic manipulator, see Fig. 1.

Fig. 1 youBot is a small versatile mobile robot platform developed for education and research. Here the manipulator arm is posed in its initial configuration

The software used is Matlab with the Simulink support for the Raspberry PI computer. To keep focus on mathematical modeling a Simulink interface for the youBot was developed. It is based on the youBot API provided with the robot.

2.1 Course Work at ICE Connections Level

The course has several labs, here two labs are presented. In the first lab students derive forward and inverse kinematics for a three linked robot. The knowledge from the first lab is later applied when they derive kinematics for the youBot manipulator arm.

2.1.1 Connections Level—Lab on Kinematics for a Three Linked Robot:

In this lab students have to assign frames and derive inverse kinematics for a planar three linked robot see Fig. 2. The robot arm has a roller attached as an end effector.

To solve the task the students need to understand the key concepts of assigning frames, and kinematic decoupling to solve the inverse kinematics. They also need to understand the key concept of how to use the Jacobian to calculate the velocity vector of the end effector. Of course the students also use the Denawit-Hartenberg convention to assign frames. The implementation in done in Matlab. After correct implementation the roller attached at the end of the robot follows the circumference of the oval shaped object, see Fig. 2. What makes this lab especially suitable is that the planar robot has kinematics that is similar to three links of the youBot manipulator arm, see Fig. 1.

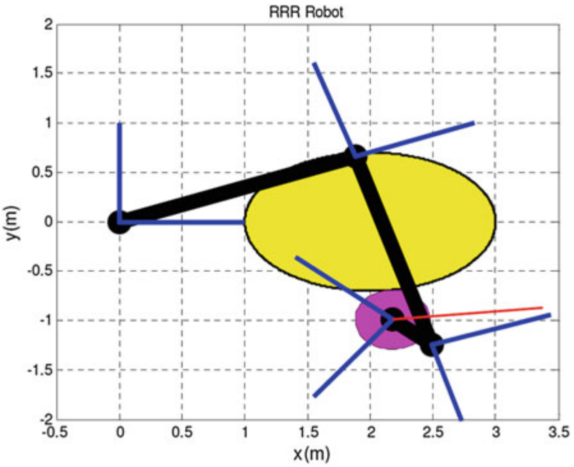


Fig. 2 A three linked planar robot with a roller represented as a *circle*, as an end-effector. The key concepts here are to calculate the forward and inverse kinematics, to assign frames, and to calculate the velocity vector of the end-effector

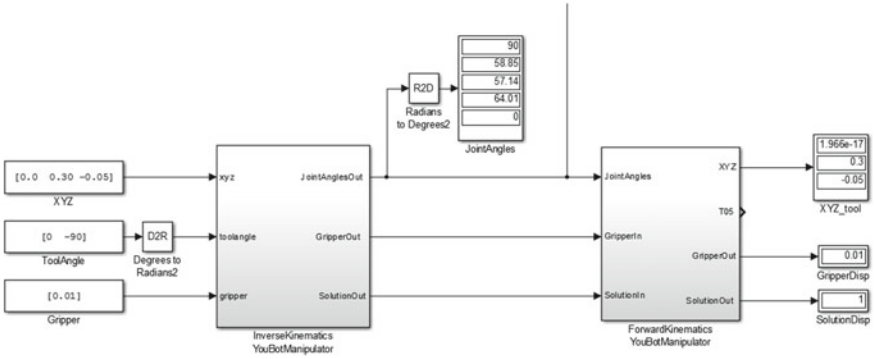


Fig. 3 The *left* block computes the inverse kinematics, and the *right* block computes the forward kinematics for the youBot. The inverse kinematics block has a boolean output that indicates if a solution for the desired pose exists

2.1.2 Connections Level—Lab on YouBot Kinematics:

In this lab students assign frames to the youBot manipulator arm according to Denavit-Hartenberg [6] convention. Then forward kinematics and inverse kinematics are derived and implemented as Simulink blocks see Fig. 3. The key concepts are forward kinematics, kinematic decoupling, and inverse kinematics for the youBot manipulator arm. From a desired pose; $[x, y, z]^T$ coordinates and gripper orientation (elbow twist and pitch angle) students need implement a block that computes the joint angles for the arm, and one that computes the forward kinematics. After implementation of inverse kinematics the students can test and validate their solutions on the real robot. An example of inverse and forward kinematics implementation is visible in Fig. 3.

2.2 Course Work at Extensions Level—Project Work

In the project students exercise what they learned at *Ideas* and *Connections levels*, they perform their understanding in a new setting. Over the year different projects have be proposed and implemented by the students. Some projects are listed here:

- Sort soft drink cans by weight. (The gripper grabs loops attached to the cans).
- youBot control using the Microsoft Kinect sensor as the operator input. The students successfully implemented and demonstrated their system which had support for lateral and transversal movement of the platform, and Cartesian movement of the gripper inside the robot workspace.
- Pick up small wooden cubes and place them on the tray (on the youBot)

Fig. 4 The youBot arm has been programmed to automatically build a tower from small wooden cubes that randomly been placed on a table in front of it. The web camera mounted on the robot arm, attached to a laptop computer, was used for the image processing. The look of the tower was designed in a GUI interface. A Raspberry Pi computer and a wireless router is visible on the tray of the robot

One recent successful student project was building a tower from small colored wooden cubes. The GUI (Graphical User Interface) was programmed in Matlab. A web camera was mounted on the gripper and was used to detect cubes and their color. Figure 4 shows a picture of the cube picking robot from when the students demonstrated their project.

2.3 Student Feedback from Course Evaluation

Here follows students feedback from year 2014 and 2015.

Student positive feedback: —*"It is very positive that we get to know how the theoretical concepts previously learnt work in a real system." (2015)*
—"i liked the amount of hours allocated to practical work and also that we were constantly supervised." (2015)
—"Working with the kuka was quite nice and challenging." (2015) —"The project was really good and challenging. I learned a lot from it. The labs were also good, they made us apply the concepts from the lecture in a real robot, what helped a lot understanding how it works." (2014)
—"The work with the YouBot was very appreciated and gave us a lot of good knowledge about how to control a real robot. The project was also a very good part of the course." (2014)

Students negative feedback: —*"The labs were efficient practice for the theoretical part of the course. But the distribution of the labs was not equal between the first and second period in the semester." (2014)*
—"I miss applying more the second part of the course in the labs. We did not really use the control part, so now it feels like it is really abstract and vague the concepts and applications of it." (2014)

> —*"The labs was mostly focused on kinematics and dynamics. The later half of the lectures was not properly covered by lab work. It would be good to have one lab with control also." (2014)*

3 Discussion

We have presented a robotics course that was revised with the ICE approach of teaching in mind.

A Simulink block that enables students to work directly on the youBot was presented. Students have successfully used them in labs and in projects.

The approach by using Simulink for code generation to Raspberry Pi has the benefit that a student can spend more time on the mathematics and modeling. From own experience I know that equal projects purely implemented in C or C++ directly on the robot itself would have consumed a lot more time.

What can be noticed in the students negative feedback is that when they did not use control part (torque control) on the youBot, the student feels it is abstract and vague. It is just an example that gives support to ICE approach of teaching.

Acknowledgments The author wants to thank the students that were involved in this study. This work was done during a revision of the Master's Programme in Robotics and Control at Umeå University.

References

1. Biggs, J., Tang, C.: Teaching for Quality Learning at University. The Society for Reseach into Higher Education (2011)
2. Entwistle, N.: Teaching for Understanding at University—Deep Approaches and Distinctive Ways of Thinking. Palgrave MacMillian (2009)
3. Fostaty Young, S.F.: Teaching, learning, and assessment in higher education: using ICE to improve student learning. In: Proceedings of the Improving Student Learning Symposium. Imperial College, London, UK. pp 105–115 (2005)
4. Perkins, D.: Chap. 2 in Teaching for understanding: linking research with practice. What is Understanding?. Jossey-Bass Publishers (1998)
5. Bischoff, R., Huggenberger, U., Prassler, E.: KUKA youBot- a mobile manipulator for research and education. In: Proceedings of IEEE International Conference on Robotics and Automation (May 2011)
6. Siciliano, B., Khatib, O.: Handbook of Robotics. Springer 2008. ISBN: 978-3-540-23957-4

Developing Extended Real and Virtual Robotics Enhancement Classes with Years 10–13

Peter Samuels and Sheila Poppa

Abstract There is growing evidence of the potential of educational robotics to enhance science, technology, engineering and mathematics education provided that they are deployed carefully. This paper describes a developmental research project between a university and a secondary school in the UK to develop extended robotics enhancement classes, mainly using LEGO MINDSTORMS robotic kits, and GeoGebra, which was used to animate virtual robots. Two styles of class were deployed: student-led project creations and facilitator-led challenges. The pedagogical principles underpinning these classes and their design are discussed. Feedback generally indicated that the classes were successful and appreciated by the students but they experienced difficulties in incorporating the virtual robotic element. Lessons learnt from the project, including the development of employability skills, the potential impact on students with autism, and the effective use of peer students, are discussed. The possibility of combining the two styles of class together is proposed.

Keyword Developmental research • Learning by design • STEM education • Robotic kits • LEGO MINDSTORMS • Virtual robotics • Geogebra • Employability skills

1 Introduction

According to Sanders [1] interest in science, technology, engineering and mathematics (STEM) education increased rapidly in the USA following the publication of Friedman's book [2] in 2005. Friedman concluded that China and India were on

---

P. Samuels (✉)
Centre for Academic Success, Birmingham City University, Birmingham, UK
e-mail: peter.samuels@bcu.ac.uk

S. Poppa
Lawrence Sheriff School, Rugby, UK
e-mail: sheila.poppa@lawrencesheriffschool.com

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_7

course to overtake the USA in the global economy by surpassing their STEM education output. The supply of employable STEM graduates has also been an increasing cause for concern in Europe in recent years [3–5]. The international comparative Relevance of Science Education research project which investigated the views of adolescent students [6] found a strong negative correlation (r = –0.85) between their interest in a future science career and the wealth of their home country. This indicates that many developed nations, including many in Europe, face the challenge of meeting the demands for STEM employment. Furthermore, inquiries into employers' views indicate that, in addition to subject knowledge, many STEM graduate roles require softer skills, such as communication, using initiative, problem solving, teamwork and creativity [7, 8] which are not traditionally taught in secondary or tertiary education.

In response to these concerns, government programs and institutional research projects have been initiated on both sides of the compulsory education threshold. Their strategies have included creating greater awareness of STEM careers through employer partnerships [7, 9] and seeking to make STEM education more enjoyable [9]. Others have encouraged the acquisition of softer skills by introducing project enhancement work into the curriculum [10] but these have mainly taken place in Higher Education due to the pressures from national and international league tables on compulsory school education [11]. These encourage schools to emphasize individual performance and teaching to test rather than promoting divergent thinking [11], preparing students for the collaborative and unpredictable world of employment, or developing a love for academic subjects, or a deeper emotional engagement with them [12].

Some initiatives to make STEM subjects more enjoyable have been criticized for lacking effectiveness [9]. However, we assert that it is not the subjects that need to be made more enjoyable, but the use of appropriate challenges in the experience of engaging with the subject that needs to be encouraged, as the subjects themselves are potentially intrinsically enjoyable to many students. The first author has described this approach as "putting the curriculum into the fun rather than the fun into the curriculum" [13, 14]. In the context of learning technologies, what is required is an understanding of which technologies are intrinsically engaging and enjoyable from the perspective of students' informal use in their free time and how such technologies could be incorporated into the curriculum [14].

One potential type of learning technology is *educational robotics*. There is growing evidence that these have the potential to enhance learning, engagement and employability skills in STEM subjects provided that they are deployed carefully [15–17]. There is also evidence of the potential of *integrated computer algebra systems* (CAS) with *dynamic geometry systems* (DGS) to enhance mathematics education when used thoughtfully [14, 18]. This paper reports on an ongoing developmental research enhancement project over the last 5 years between a secondary school and a university in the UK to develop extended robotic enhancement classes with Years 10–13 students using robotic kits in conjunction with an integrated CAS/DGS.

The background to these classes is introduced in Sect. 2. Section 3 provides a summary of the research and development framework used in the classes. Performance and feedback on the classes is reported in Sect. 4. Finally, Sect. 5 provides a discussion of the outcomes of this developmental research project as a whole and a possible improvement on the class design.

2 Background

The origins of our collaboration was a teaching idea paper [19], written by the first author and published in 2010, which suggested using robotic kits along with a mathematical simulation environment called GeoGebra (http://www.geogebra.org/) in a context of open-ended project work in order to motivate mathematics learning. The School has a selective intake and aims to develop well rounded students by blending traditional style lessons with enhancement activities, both within and outside class time. Under the UK General Teaching Council's Teacher Learner Academy (http://www.gtce.org.uk/tla/) staff were encouraged to undertake projects to stimulate their learning experiences and that of their pupils, supporting each other within and beyond their normal settings to enrich their pedagogy, thereby fostering innovation. In 2010, they were awarded LEGO Innovation Centre status [20], which enabled them to purchase LEGO MINDSTORMS robotic kits (http://www.lego.com/en-gb/mindstorms/) for use with older students, and were seeking a way to use them effectively. A member of staff from the School read the paper and contacted the first author who was given permission to work with the School.

Table 1 Summary of robotics classes

Year	Class type	Length (days)	No. students	Technologies used	Evaluation
2011	Free time, student-led projects, individual and group	5	5	LEGO MINDSTORMS, GeoGebra and Bioloid	Presentation and peer review
2012	Free time, student-led projects, individual and group	4	11	LEGO MINDSTORMS, GeoGebra and Bioloid	Presentation and peer review
2012	Enhancement class, themed challenges, group	3	17	LEGO MINDSTORMS and GeoGebra	Performance in challenges and sportsmanship
2014	Enhancement class, student-led projects, group	3	7	LEGO MINDSTORMS and GeoGebra	Presentation and peer review
2015	Enhancement class, themed challenges, group	3	11	LEGO MINDSTORMS and GeoGebra	Performance in challenges plus bonus

A summary of the classes which have been provided so far is shown in Table 1. The first two classes were slightly longer, operated in student free time and used a student-led project approach. The three later classes were organized in school enhancement lesson time; and two of them deployed a themed challenge approach.

3 Research and Development Framework

3.1 Research Methodology

The overall purpose of these classes was to *develop educational enhancement environments using real and virtual robots that are engaging, facilitate the acquisition of employability skills and motivate further STEM learning*. A research methodology appropriate for a partnership between researchers and teachers at a university and teachers and students at a school for the development of these classes was required. The nearest similar research known to the authors into the development of classes by a researcher/teacher using a similar technology is that used by Jaworski to develop classes using GeoGebra to teach algebra concepts [21]. She reflected upon her practice as a teacher in the implementation of her chosen pedagogy and concurrently as a researcher into the effectiveness of the pedagogy itself, referring to this approach as *developmental research* [22]. This paradigm was therefore chosen as it was seen to be more appropriate than analyzing and developing classes from a single role identity, such as in action research for researchers or reflection-on-action for practitioners.

3.2 Pedagogical Framework

The pedagogical framework for these classes combined several ideas from the teaching idea paper by the first author [19] with other principles elaborated in [23]. Fundamental to the former was the combined use of real and virtual robotics to motivate learning rather than teaching directly. The rationale for experimenting with this approach was the remarkable success, reported in [18], of the use of Classpad CAS/DGS calculators to motivate learning in algebra and geometry. As already explained, GeoGebra can also be used as a CAS/DGS. Using GeoGebra also provided a way to attempt to motivate mathematics learning using robotics through animations.

The *educational robotics* movement can be traced back to Papert's *Mindstorms* book [24] which encouraged social engagement and language development in a 'math' world, and which led to the (mainly virtual) turtle graphics movement. The combination of real with virtual robots is a novel idea, although it was anticipated by Burdea [25] in a wider context who foresaw one advantage being more effective *planning*. This is consistent with the use of simulations in other areas of mechanical engineering, such as computational fluid dynamics in aircraft design. Real robots are more kinesthetic than virtual ones and encourage greater social identification, which Catlin and Blamires [26] have called the principle of *embodiment*. Eisenberg [27] argued for a greater emphasis on physical robots in mathematics education as "transitional objects" which bridge the gap between concrete and formal reasoning.

Another important element of the class pedagogy was *learning by design* [28]. This promotes providing learning environments where students are given the space to create and develop their own ideas. It contrasts with *teacher-led challenges*, often involving constructing and programming a pre-planned robot design to achieve a pre-planned purpose, and *robotic competitions*, often involving pre-set challenges requiring some ingenuity. In a study of 64 engineering undergraduates, Cropley and Cropley [29] found that students without creativity training were so used to following instructions that they focused on conventional designs in robotics challenges even when they were marked for creativity. Learning by design was therefore employed as these classes were aimed at enhancing the curriculum and encouraging creativity.

The classes also made use of teamwork and peer learning. Teamwork is commonly used in robotics competitions, such as the FIRST which employs teams of 20 or more students (http://www.firstinspires.org/robotics/frc/what-is-first-robotics-competition). However, in their review of computer supported group-based learning, Strijbos et al. [30] found that teams of two or three were more effective for performing complex technical tasks due to the amount of effort required to achieve consensus. Atmatzidou and Demetriadis [31] argue that, "although the [educational robotics] practitioners have a clear orientation toward collaborative learning activities, they, nevertheless, lack a more detailed pedagogical perspective of how to tap the benefits of group-based learning". By viewing former class members as an educational resource, peer learning [32] provides such a perspective. Consistent with [11], emphasis was placed on their experience with former classes rather than their ages.

3.3 Choice of Technology

The main technologies deployed in the classes were LEGO MINDSTORMS NXT kits and the GeoGebra software environment. The first two classes also used a more sophisticated ROBOTIS Bioloid Comprehensive humanoid robotic kit (http://www.robotis.com/xe/BIOLOID_Comprehensive_en). This kit was inappropriate for the two challenge-based classes. The choice of appropriate kinds of technology for these classes is discussed in more detail in [33].

LEGO MINDSTORMS NXT kits comprise of a programmable brick which can generate sounds, LEGO bricks and other pieces, three different kinds of sensors, and servo motors [34]. They are used in conjunction with a visual programming language which controls the robot's behavior according to a series of instructions or events based on inputs received from the sensors. Once a robot has been built and a program written it can be downloaded onto the brick. The ratio of available robotic kits to students was quite high but, in order to give each team an equal opportunity, they were limited to having *two kits per team*.

GeoGebra is an open source dynamic mathematics software environment. It comprises of six alternative views covering different aspects of mathematics and statistics. In particular, it integrates a dynamic geometry system with an algebra view, enabling the representation of physical objects, such as robots, to be constructed and animated, both visually and symbolically. [35] provides the animation of a LEGO MINDSTORMS robot moving through three points on a plane which was used as the basis of a minimal instruction activity in the first three classes.

3.4 Class Design

Each class began with a series of briefing sessions on the first day led by the University partner. These all included an initial challenge to construct and program the first robot design in the LEGO MINDSTORMS NXT instructions book and a GeoGebra training session and challenge. In the more recent classes the students were organized into groups and briefed on the activities for the main period of the class. Two different styles of class were then adopted:

- *Student-led project design*: Students were given the freedom to create their own projects within the parameter of being achievable within the time period available. In the fourth class the students presented and peer assessed their plans before they created and programmed their robots. This approach was similar to that used by another UK university with their first year engineering undergraduates [10].
- *Themed challenge*s: Students were briefed on a number of specific challenges around an engaging theme for which they were required to build a robot. The third class included three challenges with an Olympic theme, coinciding with the 2012 Olympics. The fifth included a series of challenges with a Rugby theme, coinciding with the 2012 Rugby World Cup (see Fig. 1).

The middle class period lasted between one and three days and was facilitated by members of staff from the School and/or peer students who had participated in previous classes at a lower level. The final day of most of the classes started with a re-briefing session followed by a final period for robot development. This was either followed by the presentation of robot designs with a peer review of their performance in the themed challenges for which they were either scored or ranked. After this there was an award and certificate giving ceremony. Finally, students were asked to reflect upon their experiences and make suggestions for improving the classes in future.

Fig. 1 LEGO MINDSTORMS robot in the fifth class making a conversion 'kick'

The assessment of the robots for the themed challenges included a peer reviewed sportsmanship score in the third class and a peer student discretionary award for sophistication in the fifth class. These were included to encourage the students to look beyond the competitive aspects of the challenges to the wider purpose of the class.

Certificates were awarded according to the level of participation of the students:

- *Level One*: Participation in a class as an individual or group member
- *Level Two*: Facilitation of a class
- *Level Three*: Design and facilitation of a new class

Students who had been awarded a certificate were encouraged to engage in a later class at a higher level as a peer student.

4 Performance and Feedback on Classes

4.1 General Findings

Firstly, we note that these classes were held at a male secondary school with a selective intake. The students were therefore relatively intelligent, well behaved and competitive. Furthermore, in common with most UK secondary schools, the students were used to following instructions, so many found the idea of student-led project work, requiring creativity with limited rules, unusual and challenging. A surprising finding was the students' general lack of initiative with using the Internet which appeared to be due to their restricted access within their normal school environment.

Several planning meetings, sometimes involving peer students as facilitators, were held before each class took place. Students self-selected to attend classes which were advertised in the School. There were occasional issues with facilitators not fully understanding the design of the classes, which were quite different from normal lessons.

Observations and improvement recommendations were made by the university researchers, the staff facilitators, and the students as part of their reflection at the end of each class. The students participated in the training activities and generally picked up what to do quickly but some had the tendency to go off task quickly if they became bored. This confirmed the importance of the constructivist principle of seeing the training as *minimal instruction* [23].

An initial aim of the classes had been to motivate deeper mathematics learning by using GeoGebra to create accurate robotic designs requiring use of its algebra view [19]. However, the earlier GeoGebra training sessions were disappointing as some students with weaker mathematics backgrounds found the programming too hard and failed to connect virtual animation with the robotic kits. They also appeared to have a kinesthetic preference to use physical robot kits. The training was therefore simplified by only including a straight line movement, using a rectangle to represent a physical robotic table, and, moving away from a turtle graphics declarative style program, by adding a feedback event to represent an ultrasound sensor locating a wall—see Fig. 2. This was an improvement but there remained the challenge of encouraging students to use GeoGebra later in the classes. In view of these difficulties, the overall success of the robotic kits, and consistent with the developmental research paradigm, this aim was widened to motivating STEM learning in general.

Whilst the challenge-based classes were more competitive, some students lost interest once they had developed a robot to meet a challenge which initially captivated their interest. This was particularly evident in the third class when some students did not attend the full class once they had built a speed robot using gearing. However, their groups did not perform well on another more technical challenge to throw a ball. Whilst the students in the student-led classes found the freedom more daunting, they all remained engaged throughout the class. The possibility of combining these two approaches is discussed in Sect. 5.

The fifth class was mainly designed and led by two peer students from Year 13 who had participated in a previous class. They only received limited advice during their facilitation and were able to manage the sessions and keep the other students

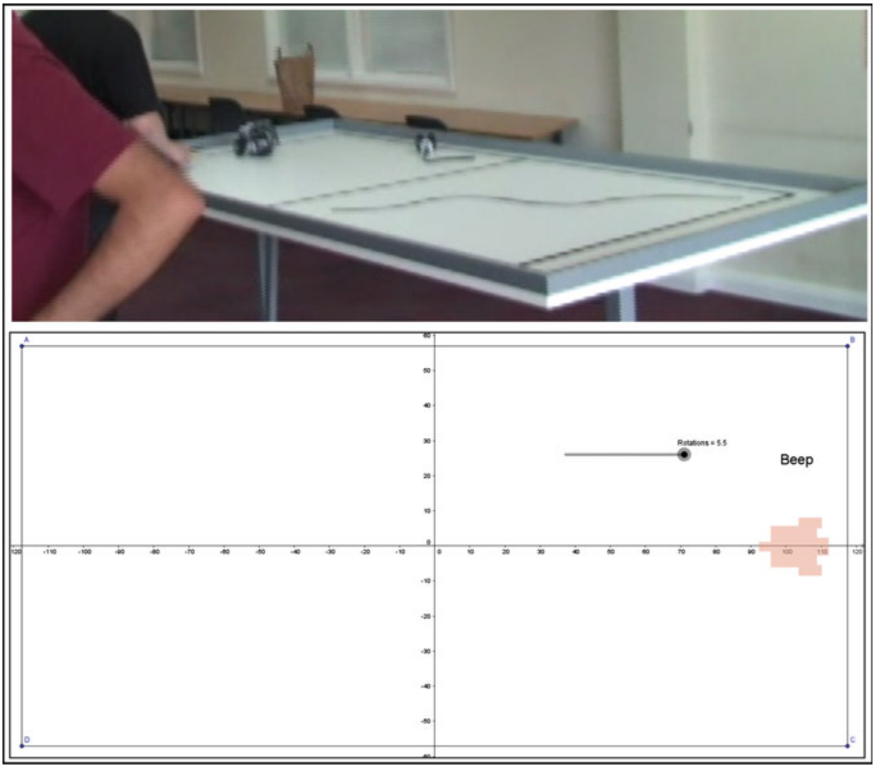


Fig. 2 Robot table and revised GeoGebra animation representing a robot translation on a table, beeping when it senses a wall (*source* https://www.geogebra.org/material/simple/id/2807751)

engaged. Whilst some of the decisions they made about rules and scoring of challenges were mildly criticized by the other students, this was seen as a positive learning experience for them to become more proficient in *design* and *facilitation*, which are themselves valuable employability skills.

4.2 Feedback

On the whole the students' feedback at the end of the classes about their experiences was positive, indicating their affective engagement [11, 12, 36]. Of the 33 written student feedbacks obtained from three of the classes 42 % mentioned enjoyment or fun whilst none of them made a negative comment about their overall experience. 12 % also mentioned that they thought the class design idea was good. Student enthusiasm was also demonstrated by some having to be told to leave at the end of the day in the school time classes whilst others gave the course leaders gifts.

In terms of the technical skills learnt, student feedback made predictable references to learning to construct (27 %) and program (36 %) robots that achieved the required goals. The LEGO visual programming language was often criticized, especially by students who had experience with other programming languages. One specific technical skill several students reported was learning how to use *gearing* (12 %) in order to make robots move faster.

Student feedback made frequent references to acquiring employability skills. The most common themes identified were *teamwork/cooperation* (79 %), *time management* (36 %) and *creativity/problem solving* (30 %), indicating that most students perceived these to be important skills that they had developed, enhancing what they had learnt from the standard School curriculum. However, there was little mention of *planning/designing* (15 %) which may have been due to the immediacy of the robotic kits encouraging repeated experimentation rather than reflection. This is discussed further in Sect. 5.

On the negative side, 53 % of students in the third class reported that the sportsmanship peer evaluation had not worked well. A reason given for this was that some teams had used tactical scoring. It was therefore decided that the facilitators should be made responsible for this in future. This appeared to work well with the fifth class as the student feedback did not comment on this aspect negatively.

4.3 Impact of Class on Students with Autism

An unexpected consequence of the first two classes was their positive impact on some autistic students. Over 10 % of students who had attended a class were identified as autistic, representing a much higher than average prevalence rate. However, all the autistic students joined in and performed well in the classes. For one student in particular the classes had a completely transformative effect to the amazement of his teachers, one of whom stating that she had “never heard him laugh before”. He was even willing to be videoed demonstrating his new-found understanding of the principle of gearing—see Fig. 3. He then progressed to facilitate a second class he attended. This success is consistent with the aspirations of [37].

It is believed that Catlin and Blamires’ principle of *embodiment* [26] is particularly relevant to autistic students as they appear to be comfortable with relating to robots as a projection of human relationships but without fear of violating social rules. Relating to other people in this context also appears to be less threatening to them, thus increasing their confidence in social engagement. Dautenhahn and Werry [38] go further in claiming that it may also be the physical movement by the autistic students themselves within the robotics class which might be therapeutic.

Fig. 3 Autistic student demonstrating to the camera the effectiveness of gearing

5 Discussion

This developmental research project has made progress towards its aim of developing an educational enhancement environment using real and virtual robots that is engaging and facilitates the acquisition of employability skills. The aim to motivate further STEM learning affected the design of the classes but was not evaluated. LEGO MINDSTORMS kits, although their software was not universally liked, have the potential to be used effectively in such classes. The use of a three level certification structure, employing peer students from previous classes as facilitators and class designers has demonstrated the value of seeing former students as an educational resource. The classes were also unexpectedly successful with autistic students, with the first two having a profound effect one student in particular.

The virtual robotic element with GeoGebra has yet to be proven, although the GeoGebra training activities have been improved. The class designers still believe that GeoGebra should be retained in order to encourage awareness and use of the employability skill of planning. Based on feedback from students in the fifth class, a possible way forward was identified. As the challenge-based activities are easier to follow but may not be as engaging it is proposed that the middle class period should be split into two halves, the first half being a series of challenges and the second half a student-led project. In order to encourage planning, as in the fourth class, students will be required to submit a plan of their project ideas for peer review before they develop and present the project itself. They will not be forced to use GeoGebra but it will be provided as an option (along with alternatives, such as animations in Microsoft PowerPoint). This new design will be investigated in the next class. The use of GeoGebra and alternatives to represent other robotic sensor events will also be explored.

The success of the project's pedagogy has also impacted positively upon the approach of the class designers in their other teaching enhancement activities, causing them to trust students more to develop their own ideas and providing them with less direct guidance. We believe this style of robotics class could be employed in similar contexts to encourage engagement with STEM education and the development of employability skills.

References

1. Sanders, M.: STEM, STEM Education, STEMmania. The Tech. Teach. **68**(4), 20–26 (2009)
2. Friedman, T.: The World is flat: a brief history of the twenty-first century. Farrar, Straus and Giroux, New York (2005)
3. Roberts, G.G.: SET for Success: The Supply of People with Science, Technology, Engineering and Mathematics Skills: The Report of Sir Gareth Roberts' Review. HM Treasury, London (2002)
4. Becker, F.S.: Why don't young people want to become engineers? Rational reasons for disappointing decisions. Eur. J. Eng. Educ. **35**(4), 349–366 (2010)
5. Henriksen, E.K., Dillon, J., Ryder, J. (eds.): Understanding Student Participation and Choice in Science and Technology Education. Springer, Dordrecht (2015)
6. Sjøberg S., Schreiner C.: The ROSE Project: An Overview and Key Findings. Technical report, University of Oslo (2010)
7. Mann A, Oldknow A.: School-industry STEM links in the UK: a report commissioned by Futurelab. Education and Employers (2012)
8. Science, Technology, Engineering, and Mathematics Network: Top 10 Employability Skills. http://www.exeter.ac.uk/ambassadors/HESTEM/resources/General/STEMNET%20Employability%20skills%20guide.pdf
9. Kudenko, I., Gras-Velázquez, À.: The Future of European STEM Workforce: What Secondary School Pupils of Europe Think About STEM Industry and Careers. In: Papadouris, N., Hadjigeorgiou, A., Constantinou, C. (eds.) Insights from Research in Science Teaching and Learning. Contributions from Science Education Research, vol. 2, pp. 223–236. Springer, Berlin (2016)
10. Adams, J., Kaczmarczyk, S., Picton, P., Demian, P.: Problem solving and creativity in engineering: conclusions of a three year project involving reusable learning objects and robots. Eng. Educ. **5**(2), 4–17 (2010)
11. Robinson, K.: RSA animate—changing education paradigms (2008). http://www.youtube.com/watch?v=zDZFcDGpL4U
12. Fredricks, J.A., Blumenfeld, P.C., Paris, A.H.: School engagement: potential of the concept, state of the evidence. Rev. Educ. Res. **74**(1), 59–109 (2004)
13. Samuels, P.C., Maitland, K.: Redefining maths learning technologies: putting the curriculum into the fun. In: 1st HEA Annual Conference on Aiming for Excellence in STEM Learning and Teaching. Higher Education Academy, London (2012)
14. Haapasalo, L., Samuels, P.C.: Five recommendations for mathematical learning technologies from the learner's perspective. Submitted to Ed. Tech. Res. & Dev
15. Melchior, A., Cohen, F., Cutter, T., Leavitt, T.: More than Robots: An Evaluation of the FIRST Robotics Competition Participant and Institutional Impacts. Heller School for Social Policy and Management, Brandeis University, Waltham, MA (2005)
16. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
17. Kandlhofer, M., Steinbauer, G.: Evaluating the impact of educational robotics on pupils' technical—and social-skills and science related attitudes. Rob. Auton. Syst. **75**, 679–685 (2016)

18. Eronen, L., Haapasalo, L.: Making mathematics through progressive technology. In: Sriraman, B., Bergsten, C., Goodchild, S., Palsdottir, G., Dahl, B., Haapasalo, L. (eds.) The First Sourcebook on Nordic Research in Mathematics Education, pp. 701–710. Information Age, Charlotte, NC (2010)
19. Samuels, P.C.: Motivating mathematics learning through an integrated technology enhanced learning environment. Int. J. Tech. Math. Educ. **17**(4), 197–203 (2010)
20. Lawrence Sheriff School: The Griffin Teaching School Alliance. http://www.lawrencesheriffschool.net/downloads-all/category/21-national-teaching-school?download=893:the-griffin-alliance-portfolio-oct-2015
21. Jaworski, B.: Challenge and support in Undergraduate Mathematics for Engineers in a GeoGebra Medium. MSOR Connect. **10**(1), 10–14 (2010)
22. Jaworski, B.: Developmental research in mathematics teaching and learning: developing learning communities based on inquiry and design. In: Liljedahl, P. (ed.) Proceedings of the 2006 Annual Meeting of the Canadian Mathematics Education Study Group, pp. 3–16. University of Calgary (2006)
23. Haapasalo, L., Samuels, P.C.: Responding to the challenges of instrumental orchestration through physical and virtual robotics. Comput. Educ. **57**(2), 1484–1492 (2011)
24. Papert, S.: Mindstorms: Children, Computers and Powerful Ideas. Basic Books, New York (1980)
25. Burdea, G.C.: Invited review: the synergy between virtual reality and robotics. IEEE Trans. Rob. Autom. **15**(3), 400–410 (1999)
26. Catlin, D., Blamires, M.: The Principles of Educational Robotic Applications (ERA): a framework for understanding and developing educational robots and their activities. In: Clayson, J.E., Kalaš, I. (eds.) Proceedings for Constructionism 2010: the 12th EuroLogo Conference, Paris (2010)
27. Eisenberg, M.: Mindstuff: educational technology beyond the computer. Convergence **9**(2), 29–53 (2003)
28. Kolodner, J.L., Crismond, D., Gray, J., Holbrook, J., Puntambekar, S.: Learning by design from theory to practice. Proc. Int. Conf. Learn. Sci. **98**, 16–22 (1998)
29. Cropley, D.H., Cropley, A.J.: Fostering creativity in engineering undergraduates. High Abil. Stud. **11**(2), 207–219 (2000)
30. Strijbos, J.W., Martens, R.L., Jochems, W.M.: Designing for interaction: six steps to designing computer-supported group-based learning. Comput. Educ. **42**(4), 403–424 (2004)
31. Atmatzidou, S., Demetriadis, S.: Evaluating the role of collaboration scripts as group guiding tools in activities of educational robotics: conclusions from three case studies. In: IEEE 12th International Conference Advanced Learning Technologies (ICALT), pp. 298–302. IEEE (2012)
32. Topping, K.J.: Trends in peer learning. Educ. Psych. **25**(6), 631–645 (2005)
33. Samuels, P.C., Haapasalo, L.: Real and virtual robotics in mathematics education at the school-university transition. Int. J. Math. Educ. Sci. Tech. **43**(3), 285–301 (2012)
34. Rinderknecht, M.: Tutorial for Programming the LEGO® MINDSTORMS™ NXT. http://www.legoengineering.com/wp-content/uploads/2013/06/download-tutorial-pdf-2.4MB.pdf
35. Samuels, P.C.: Animation of a Robot Moving through Three Points. http://www.geogebra.org/material/simple/id/2807809
36. Jones, A., Issroff, K.: Learning technologies: affective and social issues in computer-supported collaborative learning. Comput. Educ. **44**(4), 395–408 (2005)
37. Costa, S., Resende, J., Soares, F.O., Ferreira, M.J., Santos, C.P., Moreira, F.: Applications of simple robots to encourage social receptiveness of adolescents with autism. In: 31st IEEE Engineering in Medicine and Biology Society Conference, pp. 5072–5075. IEEE, Minneapolis (2009)
38. Dautenhahn, K., Werry, I.: Towards Interactive Robots in Autism Therapy: Background, Motivation and Challenges. Pragmat. Cognitive **12**(1), 1–35 (2004)

Project Oriented Approach in Educational Robotics: From Robotic Competition to Practical Appliance

Anton Yudin, Maxim Kolesnikov, Andrey Vlasov and Maria Salmina

Abstract The paper shows a way of effective organization of education and student motivation in personal digital fabrication environment. Student activity within the approach is divided in two tracks. The paper concentrates on the second, real-world project oriented track with a series of diagrams and a case study.

Keywords Educational robotics · Project-oriented education · Mobile robotics · Mechatronics · Engineering · Human capability development

1 Introduction

The current state of education in the world confronts school, industry and science with an urgent task to find effective ways of knowledge transfer to future generations taking into account the real constraints of our time. Urgent need of changes in education worldwide can and should be considered along with the global crisis in other fields of human activity: economic, political, environmental, etc. Recently emerged movement of digital fabrication laboratories spread around the world [1] could be part of the answer we are looking for. Technological advances allowed us to make another evolutionary step in industrial production which now becomes personal again [2]. Remembering the vast trade shop experience of the past it is natural to adapt the best traditions of manufacturing organization. Of course this also leads to the change in technical education.

Affordable portable electronics which appeared several decades ago allowed enthusiasts to work on numerous technical projects. This led to many breakthroughs in device design. One of those resulted in affordable and hackable 3D-printer concept widely used in production labs today. This is an example of how small groups of

---

A. Yudin (✉) · M. Kolesnikov · A. Vlasov
Bauman Moscow State Technical University, Moscow, Russia
e-mail: skycluster@gmail.com
URL: http://www.bearobot.org

M. Salmina
Lomonosov Moscow State University, Moscow, Russia

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_8

interested people can address problems big and small more efficiently and granularly and push technology forward. Such additive technology machinery uses the Planet's resources much more effectively than subtractive ones.

Though computers are widely used today their use most of the time is too far from what they are capable of or what they were meant to provide to human beings and children in particular [3]. This power to amplify human's mental abilities is still to be rediscovered for education [4] and for everyday life. Today when computers come closer to people in personal digital fabrication this step becomes obvious and brings attention of many young engineers to the problem of productive software for makers versus systems capable of reproducing old media in a new fashion.

Another way to improve the current situation with education could be in addressing real multidisciplinary team projects. Due to project learning technologies and visual methods of system design it is possible to effectively integrate real engineering tasks in educational programs [5, 6].

The paper shows a way of effective organization of education and student motivation in personal digital fabrication laboratory environment. The educational process being reviewed considers continuous progressive consistent approach complementing classic elementary school, secondary school and university, uniting and extending knowledge obtained within these common means. Results of such an educational process are presented with a case study.

2 Project Oriented Approach to Education

The authors present supplementary education approach and emphasize the possibility of pairing knowledge obtained in ordinary nowadays' schools with practice of actual real-world device design.

As shown in Fig. 1 the main track to gain a competence level for a newcomer is through competition projects. Mobile robot competition is a tool for basic project-based engineering education. Tasks for students are drawn from the competition's rules. The choice of a mobile robot as an object for study is due to possible project difficulty versatility.

Competition projects' track could be described as intensive training for intellectual sport's competition event. Appropriate age for a student to start a normal study process for this track is 10 years old. For expected results the minimum work pace is 2 study sessions per week (each session is 2 h long).

Junior student start age is around 8 years old. Juniors would normally double the time needed to take the competition projects' track before they are ready for individual work resulting in 2–4 years of such studies.

After a year or sometimes two a student is able to form his own project ideas and be involved in parallel individual project's track. This leads to more working hours and higher overall competence achieved. The projects results often carry economical value but still show room for education continuation within the digital fabrication laboratory environment.

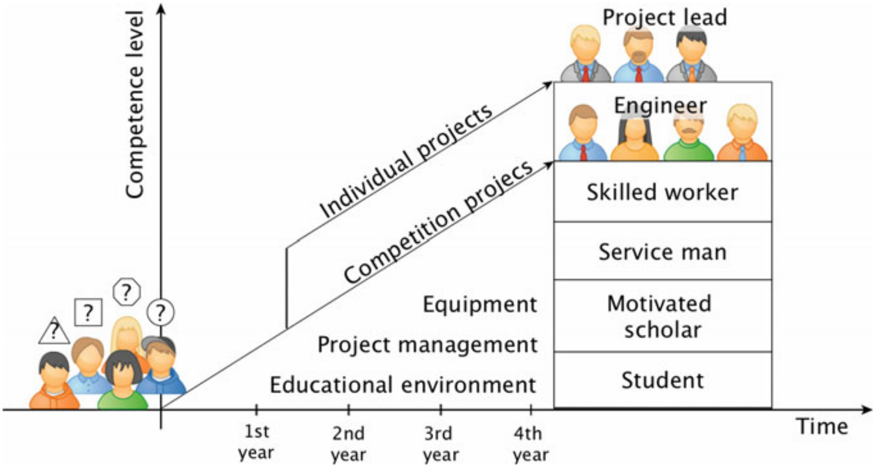


Fig. 1 Project oriented approach to education

Individual projects' track could be described as real-world project training with all the limitations, deadlines and project management seen in industry's everyday work. The minimum work pace is 2 study sessions per week (each session is 2 h long). These sessions are independent of other tracks' sessions meaning their time adds up to the competition track study time.

It is important to mention that both tracks form enough knowledge for a student to start off his career even without finishing the expected 4 years of studies. As shown in Fig. 1 after approximately 2 years he is able to work with machinery as a service man or consult customers at an electronic components' store. After 3 years he is able to organize simple work-flows and work with simple technical designs. After 4 years of practice he is a qualified engineer able to solve complicated technical cases and even lead such projects if he was hard-working enough in studies.

It is important to mention that by "a qualified engineer" authors mean practical aspects of the profession. The student is able to understand how to solve complicated complex technical tasks, see the direction to take to actually succeed in development but of course he still lacks university knowledge of in-depth subjects to do that scientifically correct and effective. Simply put, the student by himself is able to choose the right university and the technical area for his further studies and is ready to enter the chosen institution at the moment of successful graduation from the discussed course.

The competition projects' track was described previously by the authors [7–9]. There is no need to repeat that work as the interested reader could always address the mentioned work.

Figure 2 shows a typical individual project's evolution in the described context and study subjects required or involved in the process. Projects in this track are based on robotics which is multidisciplinary in nature. But since the developed projects'

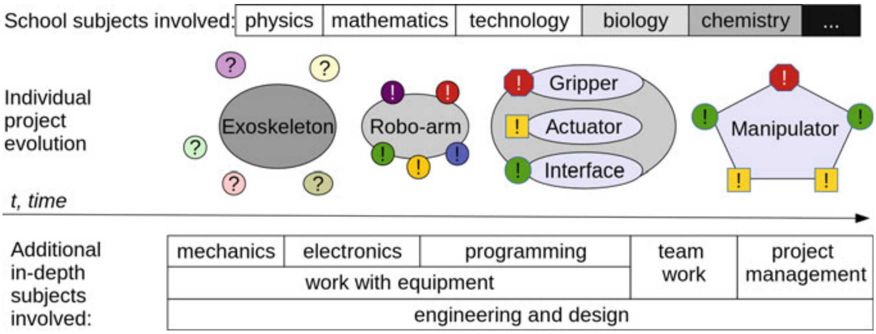


Fig. 2 Typical individual project's evolution

ideas are proposed by students and are inspired by the real world problems technical disciplines are usually expanded with other sciences as well.

In this case we have an idea of making an exoskeleton which comes from students. It has a lot of questions at first which might not be clear to those who propose. On the next step it is discussed with a teacher and the core idea changes to "robot arm" as it is a more adequate proposition in the given technological environment, resource constraints and deadlines. Next step of decomposition gives each projects' team member an individual task to perform in the given time. Some of those tasks are design oriented, engineering ones and some are scientific. Finally at some predefined time parts of the projects are assembled together to form a product. The cycle could continue from decomposition stage to find better solutions or to contribute to a bigger project goal.

In the presented particular case the time span for the project is 3 mon. In the end of the project students could present a working gripper with several human-machine interfaces. While the usual actuator worked fine (servo) the scientific proposition for a new type of magnetic fluid based actuator was not ready and needed more time to "translate" the "invented" physics principle to an engineered device. This could become a goal for the next cycle of the project.

3 Approach Results and Practical Appliances

A real-world project developed by those who successfully finished the proposed mobile robot competition training track will be presented below. The idea behind this case study is to show how technical systems developed in a competition based educational project can be used in further student work (which could be inside the individual projects' track or even outside the educational environment). This way of thinking about their studies allows students to form united teams with a technological platform able to cope with many real-life tasks for starting their after school career or business.

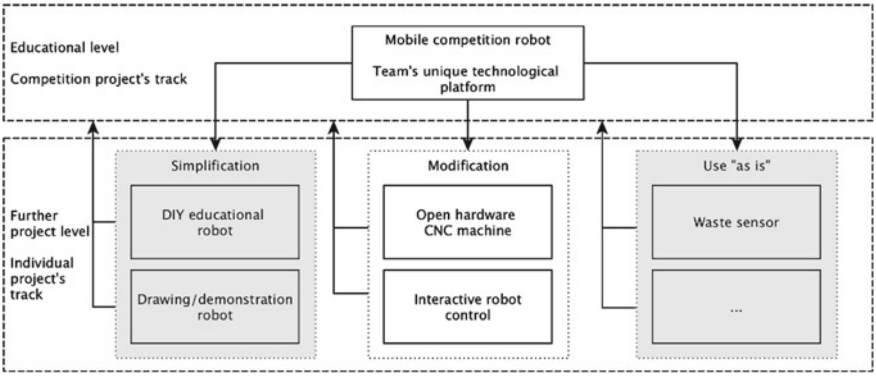


Fig. 3 Applying competition track's results to real-world problems

Figure 3 shows possible cases divided into categories like "Simplification", "Modification" and "Use as is".

"Simplification" category gathers projects which exploit mobile competition robot's systems developed by a team in the competition project's track. Here the robot's tasks are practical but simple or carry additional educational value. Such projects play an important role in forming the right educational environment when students teach each other.

"Modification" category gathers projects which bring evolution to mobile competition robot's systems developed by a team in the competition project's track. Such projects allow getting a new perspective on how the robot's systems could be transformed in something useful but different from the logical point of view. An example of such a case could be a CNC machine, using same motors and control electronics, but requiring different mechanics and programming techniques. Such projects usually add something new to the team's unique technological platform.

"Use as is" category gathers projects which use more or less same systems from robots developed by a team in the competition project's track. Such projects show how "powerful" the team's technological platform is in real-life applications. These projects of course could involve slight modifications but generally they are relatively quick to develop compared to starting from scratch. Such a project we'll discuss in detail below.

Figure 4 shows typical mobile autonomous robots that participated to an international competition.

The figure is flagged with sensor positions. Both robots use ultrasonic range finders in their obstacle avoidance systems to track mobile opponent's position and other static field objects. This is a common competitions task that is solved in a similar way by many participating teams.

Getting to know the sensor part of robotic systems in the competition track which like in the presented case are ultrasonic based or any other part as well could lead to

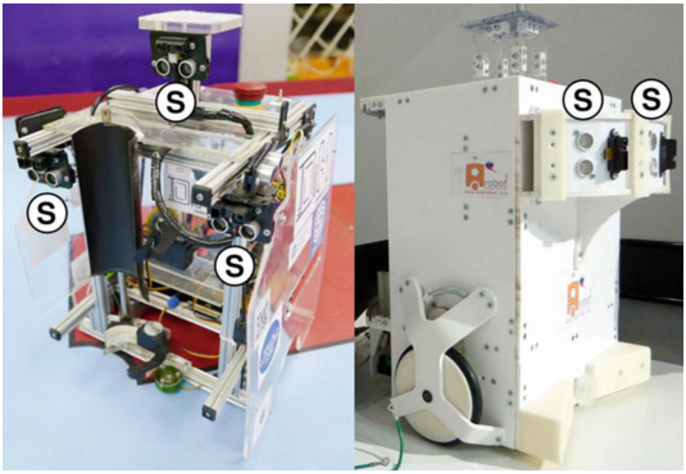


Fig. 4 Typical mobile autonomous robots designed in a competition track

many real-world applications using the same physics. An example of such an application is presented below as a case study.

4 Case Study: Waste Sensor

Garbage handling and collection is a big problem in most of the cities on our planet. When it seems that everything around us has become digital - garbage is still neglected on the global scale for some reason. Though existing electronics and sensors show a very good potential to help in solving the problem of effective garbage collection and saving valuable resources while optimizing the collection routes for example. One of the typical pictures depicting the problem is shown in Fig. 5.

Building an autonomous range finder sensor with a wireless connection to Internet, durable enough to work for several years without maintenance and cheap in production could help in this case (Fig. 6). Such a sensor is placed in every garbage bin forming a network of wireless devices. Each sensor sends its status (full or empty) to the central server where all the routes for garbage collection are computed automatically.

There are 2 design variants for the sensor unit: for mounting it on the garbage bin lid and for mounting it on its wall.

This project was based on prior competition robot's building experience. The ultrasonic sensor was chosen for the range finder role. This system naturally comes out of the robots presented previously in Fig. 4.

Fig. 5 Overfilled garbage bins in a street of Paris

Fig. 6 Autonomous garbage level sensor unit—2 design variants

The sensor unit consists of two main parts: a body containing all control electronics with a battery unit (on the Fig. 7 to the right) and a base used to firmly attach the sensor to the garbage bin (on the Fig. 7 to the left).

Fig. 7 Sensor unit composition

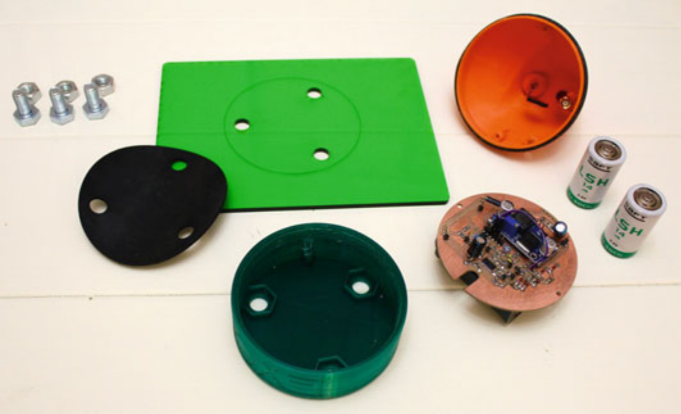


Fig. 8 Sensor unit detailed composition

Dimensions of the unit are determined by the size of the internal components: the PCB with control electronics and communication antenna's height (Fig. 8). Ultrasonic range-finder is placed with force fitting in the top part of the body in a special deepening. The base has 3 holes for bolt fixing it to the bin. For better hermetic sealing between the body and the base a rubber ring can be placed as well as a rubber pad between the base and the garbage bin.

The sensor body's form has no corners or protruding parts and it was chosen based on heavy usage condition in a garbage bin (the sensor must not be damaged or knocked down, and must not prevent garbage from its normal movement during filling/emptying the bin). The PCB is fixed to the body with 2 screws.

The base allows to firmly attach the entire sensor to the bin and at the same time serves as a lid covering the hole in the body and hermetically sealing it.

Mounting of the sensor body to its base occurs by means of a threaded connection.

A special locking screw (Fig. 9) is introduced in the bottom of the body in order to prevent undesirable unwinding of the sensor's parts (accidental or intentional).

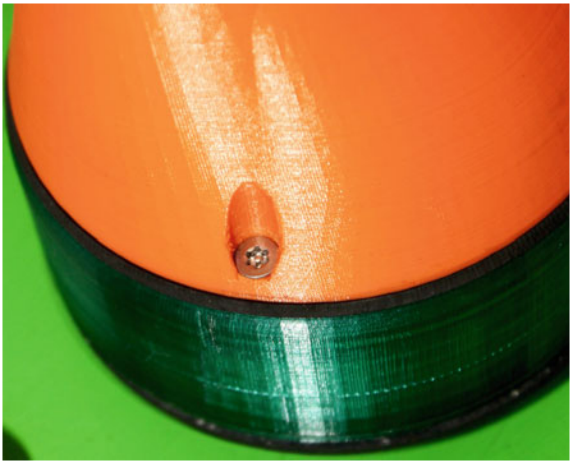


Fig. 9 Locking screw in the sensor body to prevent unwinding

The sensor unit fabrication is a complex process. In the presented case it was simplified to match those main machines which are in possession in a digital fabrication laboratory of the development team: 3D-printer, laser cutter, precise milling machine. Finally the unit body and base are meant to be produced with moulding and casting.

3D-printing requires a computer modelling step. The sensor base and body are developed within this stage. It is important to consider technological constraints of both 3D-printing hardware and later moulding and casting process.

Moulding and casting could be an expensive operation and that is why it is important to separate it from the better form search which could be done easily on a 3D-printer. In the presented case there were 4 stages of the sensor unit's outlook evolution (see Fig. 10).

While 3D-printing is a convenient means of prototyping it is important to consider the time needed to produce the print. For the sensor body in this project it takes about 21 h (Fig. 11), the base—14 h. This makes the printing to be on the critical path of the project realization.

Electronics is prototyped on a precise milling machine called modella. The process of electronics development is carried out in parallel to other processes of the sensor's fabrication. In this case electronics design also went through a 4-stage evolution (3 first stages are shown in Fig. 12). The final stage result is meant to be produced in the classic way with a factory order.

The development process took about 3 mon to finish the prototype for the customer. Most of the work was done in the field of 3D-printing (modelling and fabrication) and in the field of electronics design to meet the requirements. Programming was one of the steps which happened easier than others as it was very similar to prior robotic applications built by the team.

While most of the work (like the electronics design and programming) was made by senior students and graduates of the discussed course, the project was successfully

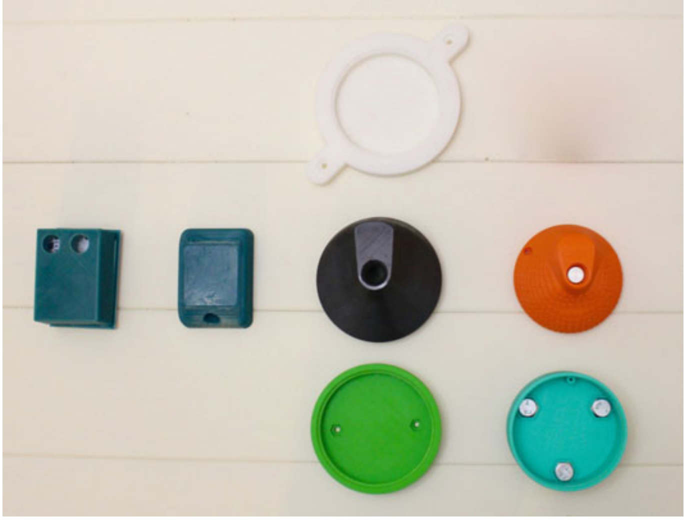


Fig. 10 The sensor unit outlook evolution

chine
zle size (mm) 0.4

uality
Initial layer thickness (mm) 0.3
Initial layer line width (%) 100
Cut off object bottom (mm) 0.0
Dual extrusion overlap (mm) 0.15

Speed
Travel speed (mm/s) 150.0
Bottom layer speed (mm/s) 20
Infill speed (mm/s) 75
Top/bottom speed (mm/s) 40
Outer shell speed (mm/s) 40
Inner shell speed (mm/s) 60

Cool
Minimal layer time (sec) 5
Enable cooling fan ☑

20 hours 18 minutes
8.31 meter 66 gram

Fig. 11 3D-printing parameters and technological restrictions

used in the educational process for all groups of students in the developing educational lab. A lot of original ideas like the "rotational" design of the case and ability to measure with ultrasonic sensor were at first proposed and tested by 2nd and 3rd year students and later adapted to the project's needs by the core developing team of seniors.

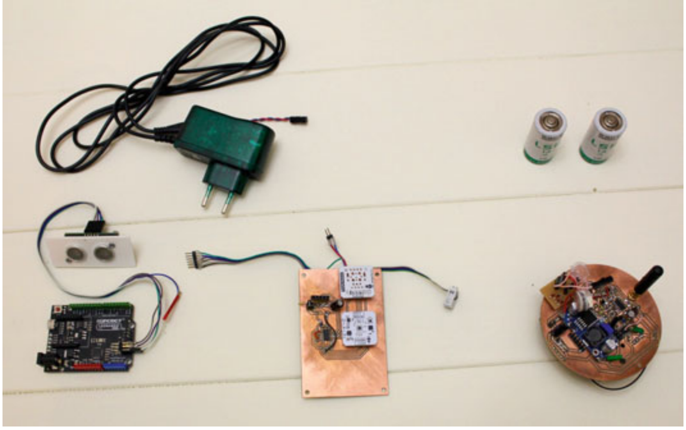


Fig. 12 Electronics design evolution

5 Conclusion

The paper summarizes and structures information on project oriented approach to technical education which authors obtained in more than a ten-year period of skilled engineer preparation within robotic competition teams. Some of the recent ideas and constructs are included in the work to make another step towards the description of a working approach to education.

Student activity within the approach is divided in two tracks, the initial motivation and main educational track is based on robotic competitions and thus is called the competition projects' track. The complementary competence developing track is based on real-world application projects with economical value and possibility to work with real customers. It is thus called the individual projects' track.

Most of the paper is describing the individual projects' track. In the competition projects' track the time is measured in years because each educational cycle to prepare a robot for the competition takes 1 school year. In the individual projects' track the time is measured in periods of 3 mon which is a comfortable time scale for a customer to wait for a desired prototype of his or her system.

Approach results and a typical system's demonstration that could be easily derived from the experience of building a competition robot are presented with a case study of building a waste sensor for a commercial customer. Some of the other possible uses of such a unique team's technological platform are presented with a category chart.

The paper fills another part of methodology description the authors started several years ago and could be interesting to those seeking practical techniques of technical education and engineer preparation in the emerging new world of widely spread personal digital fabrication.

References

1. Gershenfeld, N.: How to make almost anything, the digital fabrication revolution, Foreign Aff. **91**(6) 2012. http://www.foreignaffairs.com/articles/138154/neil-gershenfeld/how-to-make-almost-anything
2. Gershenfeld, N.: The third digital revolution, presented at the Solid 2014 Conference, San Francisco, CA, USA, May 2122, 2014. http://solidcon.com/solid2014/public/schedule/detail/35425
3. Kay, A.: The Real Computer Revolution Hasnt Happened Yet, Viewpoints Research Institute, 1209 Grand Central Avenue, Glendale, CA 91201, 2007. http://www.vpri.org/pdf/m2007007a_revolution.pdf
4. Kay, A.: Thoughts About Teaching Science and Mathematics To Young Children, Viewpoints Research Institute, 1209 Grand Central Avenue, Glendale, CA 91201, 2007. http://www.vpri.org/pdf/m2007003a_thoughts.pdf
5. Shakhnov, V., Vlasov, A., Zinchenko, L., Rezchikova, E.: Visual learning environment in electronic engineering education. In: 2013 International Conference on Interactive Collaborative Learning (ICL), Proceedings (2013)
6. Okunev, Y., Dovbysh, S., Lokshin, B., Salmina, M., Formalskii, A.: Programme scientifique et educatif pour des eleves et des instituteurs de mecanique, de mecatronique et de robotique. In: 9 Colloque Francophone de Robotique Pedagogique, ser. Pre-Actes, pp.141–143. La Ferte-Bernard, France, 14–16 May 2007
7. Yudin, A., Sukhotskiy, D.: Startup robotics course for elementary school. In: Research and Education in Robotics—EUROBOT 2010, ser. Communications in Computer and Information Science, vol.156. Rapperswil-Jona, pp.141–148. Springer Berlin Heidelberg, Switzerland, 27–30 May 2010
8. Yudin, A., Sukhotskiy, D., Salmina, M.: Practical mechatronics: training for mobile robot competition. In: Dessimoz J.D., Balogh R., Obdzralek D. (eds) “Robotics in Education, RiE 2015”, Proceedings of the 6th International Conference on Robotics in Education, RiE 2015, HESSO.HEIG-VD, Yverdon-les-Bains, Switzerland, 20-23 May, 2015. Roboptics Editions, pp.94–99. Cheseaux-Noreaz, Switzerland (2016). ISBN 978-2-9700629-5-0
9. Salmina, M., Kuznetsov, V., Poduraev, Y., Yudin, A., Vlasov, A., Sukhotskiy, V., Tsibulin, Y.: Continuous engineering education based on mechatronics and digital fabrication. In: Dessimoz J.D., Balogh R., Obdzralek D. (eds) “Robotics in Education, RiE 2015”, Proceedings of the 6th International Conference on Robotics in Education, RiE 2015, HESSO.HEIG-VD, Yverdon-les-Bains, Switzerland, 20–23 May, 2015. Roboptics Editions, pp.56–57. Cheseaux-Noreaz, Switzerland (2016). ISBN 978-2-9700629-5-0

ER4STEM Educational Robotics for Science, Technology, Engineering and Mathematics

**Lara Lammer, Wilfried Lepuschitz, Chronis Kynigos,
Angele Giuliano and Carina Girvan**

Abstract Robotics is a popular vehicle to introduce young people to science, technology, engineering and mathematics (STEM) with various approaches worldwide that use robotics to teach or entertain or both, accompanied by various tools and repositories. However, the stakeholders involved have different goals and methods, thus difficulties in finding common ground. E.g. the focus in most cases is on increasing interest in STEM, but research methods are unspecified or vague; or despite the vastness of offerings, teachers are reluctant to incorporate activities in the classroom. In this paper, we introduce the Educational Robotics for STEM (ER4STEM) project that will realize a creative and critical use of educational robotics to maintain children's curiosity in the world. An open and conceptual framework will bring three main stakeholders of educational robotics—teachers, educational researchers and organizations offering educational robotics—together through a user- and activity centered repository.

Keywords Educational robotics · Framework · Teachers · Repository

---

L. Lammer (✉)
ACIN Institute of Automation and Control, Vienna University of Technology, Vienna, Austria
e-mail: lammer@acin.tuwien.ac.at

W. Lepuschitz
PRIA Practical Robotics Institute, Vienna, Austria
e-mail: lepuschitz@pria.at

C. Kynigos
Educational Technology Lab, University of Athens, Athens, Greece
e-mail: kynigos@ppp.uoa.gr

A. Giuliano
AcrossLimits Limited, Hamrun, Malta
e-mail: angele@acrosslimits.com

C. Girvan
Cardiff University, Cardiff, Wales, UK
e-mail: girvanc@cardiff.ac.uk

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_9

1 Introduction

Robotics is a popular vehicle to introduce young people to science, technology, engineering and mathematics (STEM) with various approaches worldwide. The most common aims of these activities are either the teaching of robotics and STEM subjects or invoking young people's interest in STEM fields and careers [1]. Consequently, various stakeholders are involved; e.g. teachers looking for interesting ways to teach their subjects, researchers developing pedagogically informed activities to foster certain skills, robotics enthusiasts offering fun and play, or educational technology developers commercializing different tools.

In the current situation in Europe, the difficulty presents itself not so much in the quality of the different offerings but rather on, firstly, evaluating their impact, and secondly, bringing different stakeholders together for synergies. As a consequence, the main stakeholder—the young people—is lost out of sight; e.g. many activities engage already interested young learners but fail to attract others, technology has become more important than the learning activity or educational robotics schedules are fragmented and touch young learners only once if at all (e.g. [2, 3]). There is a pressing need to exploit the multidisciplinary potential of robotics with a well structured interdisciplinary approach to foster children's curiosity, leading them to entrepreneurial, industrial and research careers in STEM fields.

2 Educational Robotics

Robotics is an excellent tool for teaching science and technology [4], therefore many educational robotics activities focus on STEM [1]. One of the most successful and long lasting initiatives to promote STEM in Europe, especially focusing on girls, is the Roberta Initiative [5] that uses special gender-appropriate teaching and learning materials. Besides these, the long-term success of the initiative may be attributed to the certification process of "Roberta teachers" and the repository with access to a wide variety of materials. The main goal of the project is to engage and motivate girls and boys to take a sustained long-term interest in STEM.

There are two other interesting former European projects. One is TERECOP (Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods) with the aim to develop a framework for teacher education courses in order to enable teachers to implement the robotics-enhanced constructivist learning in school classrooms, and report experiences from the implementation of this framework [6]. This framework can be very helpful in designing activity plans and new curriculums that enhance STEM and Educational Robotics education. The other is Centrobot [7], with the aim to stimulate the interest of young people in technology and research. The Centrobot project organized international competitions in the Vienna-Bratislava region, two scientific conferences as well as three summer schools and an exchange program within the educational sector. In that way students engaged in robotics, learned about STEM and got involved in activities with students from other countries.

Robotics competitions (e.g. First League or RoboCup Junior) are one of the most widespread events regarding educational robotics activities. They offer a structure for problem solving in group settings by encouraging focused hands-on problem solving, team work, and innovation [8]. They are also social events where young people from different countries meet each other. However, the focus is mostly on technical problem solving, thus, certain types of students and teachers are attracted to participate. Robotics workshops are one of the other most widespread activities, maybe more diverse, addressing young people according to age, either in or outside school settings. Some activities follow a curriculum; others are rather summer camp activities. The main purpose is either the teaching of robotics or STEM subjects and invoking young people's interest in STEM fields and careers, which is followed and documented more or less systematically depending on the organizer of the activity.

The above mentioned projects and initiatives in Europe involve different stakeholders and address different aspects in educational robotics (Fig. 1). Young people and teachers are often addressed as if each was a homogenous group, which they are not. E.g. not all young people may be competitive or have talents in STEM fields; or not all teachers may be talented tinkerers willing to read through tutorials in repositories of technology tools. Organizers of educational robotics activities are also a very heterogenous group with different backgrounds and motivations, e.g. some may use robots as a motivational factor to entertain children, and the teaching or invoking interest in STEM may be a by-product. On the other hand, educational researchers may rather be interested in developing pedagogically informed activities and measuring the impact in an empirical way, e.g. improvement of skills or change of science related attitudes [9]. A closer look at the educational robotics landscape reveals that the different stakeholder needs and requirements should be addressed in a more structured way, and brought together under one framework to leverage synergies.

Fig. 1 Educational robotics stakeholders

3 The ER4STEM Project

The ER4STEM project sets out to create a continuous STEM schedule by leveraging on different already existing European approaches of innovative science education methods and measures based on robotics within one open operational and conceptual framework. Students aged 7–18 as well as their educators will be offered different perspectives and approaches to find their interests and strengths to pursue STEM education or careers through robotics and semi-autonomous smart devices. At the same time students will learn about technology (e.g. circuits), about a domain (e.g. math, physics, biology, psychology) and acquire skills (e.g. collaborating). New methods will be developed to achieve an integrated and consistent concept that picks children up at different age levels starting with primary school to accompany them until graduation from secondary school.

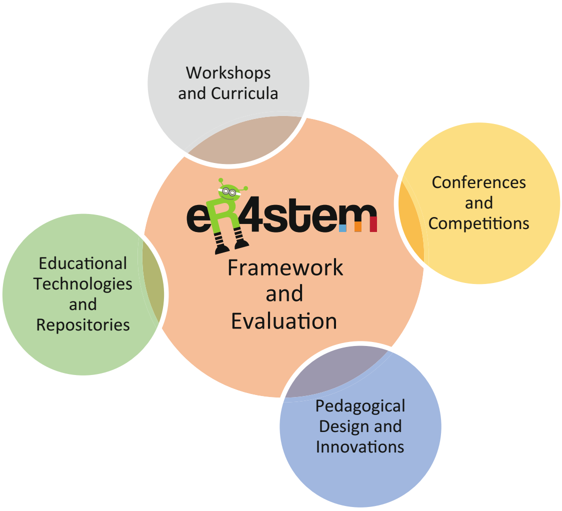
ER4STEM Framework

The ER4STEM framework builds the base of all activities and innovations offered. It will create processes, tools and artefacts that allow the use of robots in learning spaces and will be the catalyst to improve young people's learning experience through the use of robotics in formal and informal spaces. Different perspectives inform the framework (Fig. 2): Workshops and curricula, conferences and competitions, pedagogical design and innovations, and educational technologies and repositories. The whole concept will undergo a rigorous evaluation.

Workshops and Curricula

Each partner develops specific content and organizes workshops each year that provide multiple-entry points and facilitate a continuous STEM schedule. The workshops cover three different age groups (7–10, 11–14 and 15–18) and will be

Fig. 2 ER4STEM structure



offered as units in schools or informally to be attended by young people during their school-free time. More than 4000 students will be reached by the workshops. The evaluation of these activities will flow into the framework and inform the design of the generic process for development of workshops and curricula that empower children.

Conferences and Competitions

Conferences designed with and for young people provide an opportunity to present theoretical work and demonstrate artefacts to exchange ideas with each other and meet robotics researchers and companies. The competition aspect completes these activities by not only focusing on the best and quickest robots built by the student teams, but also on the provided documentation and the quality of technical solutions. The evaluation of the ECER[1] conference will flow into the framework and inform the design of conference and competition development.

Pedagogical Design and Innovations

There is a need to describe educational robotics activities analytically to become more explicit and elaborate about pedagogical design and also have activities that can be shared and compared. Parameters and criteria need to be defined to be able to identify best practices and innovations in pedagogical activities. Tools to help become more explicit and elaborate about pedagogical design will be developed and tested.

Educational Robotics Technologies and Repositories

A user-centred repository will reflect the framework and focus on mapping the processes needed by its stakeholders. The aim of bringing "added value" for the users will help avoiding the common trap of "another great resource that is left unutilized" and ensure sustainability after the end of the project. The repository will be populated with a variety of open content made of different types of learning objects and materials, links to outside resources, and educational tools built during the project (Hedgehog and Slurtles prototypes). In order to do this, the seven principal purposes of Open Educational Ressource (OER) repositories [10] including quality, flexibility, ease of use and collaboration will be followed.

Evaluation

Both quantitative and qualitative data are collected across studies in educational robotics. While qualitative data is collected in large scale studies (e.g. [11]), these studies often lack rigorous and systematic analysis of the data and can become anecdotal. There is little triangulation between data sources to increase reliability and validity of claims made. In smaller scale studies there is evidence of qualitative data analysis addressing this issue, e.g. [12]. The challenges of ER4STEM evaluation will be to undertake high-quality analysis of qualitative data at scale to evaluate the impact of the framework tools and activities on young people.

---

[1] http://pria.at/en/ecer.

4 Conclusion

This paper introduced ER4STEM, a project that aims to realize a creative and critical use of educational robotics to maintain children's curiosity in the world leading them to entrepreneurial, industrial and research careers in STEM fields, by exploiting the multidisciplinary potential of robotics with a well structured interdisciplinary approach. The project will provide an open and conceptual framework with processes, tools, and artefacts for educational robotics stakeholders to find common grounds and ease collaboration between each other. The underlying principles will be mapped in a user- and activity-centered repository that follows the seven principals of Open Educational Resources [10].

The main stakeholders of the framework and repository have been identified already: teachers, educational researchers and organizations offering educational robotics activities. They are involved in the design process from the beginning to the end. Their needs and requirements will build the use cases on which the framework and repository will be designed and tested. The workshops and conferences have also started in many project consortium countries. They have been described with pedagogically informed activity plans and their evaluation results will also inform the framework and repository development.

Acknowledgments The project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 665972. We would like to thank our partners in ER4STEM project European Software Institute CEE and Certicon and all other contributing members of the ER4STEM team.

References

1. Baretto, F., Benitti, V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
2. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
3. Bredenfeld, A.H., Steinbauer, G.: Robotics in education initiatives in Europe—status, shortcomings and open questions. In: Teaching Robotics-Teaching with Robotics—SIMPAR Workshop, Darmstadt (2010)
4. Mataric, M.J.: Robotics education for all ages. In: Proceedings of AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education, Palo Alto (2004)
5. Bredenfeld, A.H., Leimbach, T.: The roberta initiative. In: Proceedings of SIMPAR 2010 Workshops International Conference on Simulation, Modeling and Programming for Autonomous Robots, Darmstadt (2010)
6. Alimisis, D.: Robotic technologies as vehicles of new ways of thinking, about constructivist teaching and learning: the TERECoP project. IEEE Robot. Autom. Mag. **16**(3), 21–23 (2009)
7. Balogh, R., Dabrowski, A., Hammerl, W., Hofmann, E., Petrovič, P., Rajníček, J.: Centrobot Portal for Robotics Educational Course Material (2011)
8. Feil-Seifer, D., Mataric, M. J.: Defining socially assistive robotics. In: International Conference on Rehabilitation Robotics (2005)

9. Kandlhofer, M., Steinbauer, G.: Evaluating the impact of educational robotics on pupils' technical- and social-skills and science related attitudes. Robot. Auton. Syst. **75** Part B, 679–685 (2016)
10. Robinson, M.: Robotics-driven activities: Can they improve middle school science learning? Bull. Sci. Technol. Soc. **25**(1), 73–84 (2005)
11. Somyürek, S.: An effective educational tool: construction kits for fun and meaningful learning. Int. J. Technol. Des. Educ. **25**(1), 25–41 (2015)
12. Jung, I., Sasaki, T., Latchem, C.: A framework for assessing firness for purpose in open educational resources. Int. J. Educ. Technol. Higher Educ. **13**, 3 (2016)

Part III

Design and Analysis of Learning Environments

The Educational Robotics Landscape Exploring Common Ground and Contact Points

Lara Lammer, Markus Vincze, Martin Kandlhofer and Gerald Steinbauer

Abstract In the last decades, educational robotics has gained increased attention evoking a need to discuss and document different approaches and lessons learned. In this article, we report our findings made during the "Educational Robotics Café", a workshop format where experts engage in an open discussion about opportunities and challenges of the educational robotics landscape as well as advantages and shortcomings of various approaches. Interestingly, participants working on different educational robotics topics with different methods realized that all seemed to have similar problems and experiences. They could define areas of common ground, yet had difficulties in finding contact points between their educational robotics approaches to compare them. Known categorizations seemed not to fit or to be too high level. Based on these findings, we finish our article by suggesting a "tagging" approach to enable better communication between experts from different domains like education or robotics.

Keywords Educational robotics · Teacher · School · Society

1 Introduction

In the last decades educational robotics has gained increased importance and attention worldwide [1]. Many different educational robotics approaches and frameworks exist [2], yet, teachers and educators face the problem of keeping an overview and

---

L. Lammer (✉) · M. Vincze
Automation and Control Institute, Vienna University of Technology, Vienna, Austria
e-mail: lammer@acin.tuwien.ac.at

M. Vincze
e-mail: vincze@acin.tuwien.ac.at

M. Kandlhofer · G. Steinbauer
Institute for Software Technology, Graz University of Technology, Graz, Austria
e-mail: kandlhofer@ist.tugraz.at

G. Steinbauer
e-mail: steinbauer@ist.tugraz.at

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_10

finding the right ones for their needs. Consequently, it is important to discuss and document concepts as well as lessons learned and failure stories. In order to stimulate a vital discussion process among different researchers, teachers and experts in the field of educational robotics, we conceptualized and conducted a workshop[1] guided by a simple question: "*Which approaches do actually work when teaching children robotics?*"

2 Educational Robotics Café

2.1 *Method*

To provide structure and guide discussions, we chose the combination of two methodologies. One is the World Café, a methodology of collaborative inquiry and learning with the aim to create knowledge. The café metaphor is used to bring the focus to a space where dialogue, reflections and shared meaning—or "conversations that matter"—happen [3]. The other is SWOT analysis, a strategic planning tool to evaluate strengths and weaknesses, opportunities and threats. The resulting SWOT matrix contrasts the results of the internal analysis (strengths and weakness) and the external analysis (opportunities and threats) to define strategic fields of action [4]. The goal was to encourage a dialogue between different experts to create the educational robotics "landscape", first by mapping the environmental factors that either pose opportunities or threats, then by placing different approaches and methods on it, highlighting advantages and shortcomings. For instance when teachers' fear of complex technology poses a "threat" to educational robotics, it can be addressed for example with plug and play technology, which can be simple in use (strength) but expensive (weakness).

2.2 *Procedure*

Participants and Setting: In sum 13 experts from different fields concerning robotics (researchers, teachers and students) participated in the workshop. Participants were divided on four tables, where we tried to create a relaxed atmosphere with different tools to write and doodle. Coffee and sweets were in another room but could be taken to the tables. The workshop lasted three hours plus half an hour break.

Pre-phase: Before the cafe discussions started, four researchers presented their approaches to the audience that were reflecting different angles of the same goal: "teaching children with robots". While the approach A [5] concentrated on teaching

---

[1] At the 6th International Conference on Robotics in Education (May 2015, Yverdon-les-Bains, Switzerland).

natural science to primary school children in a holistic way by using robotics as a tool, the approach B [6] brought together children from secondary school with children from kindergarten and their grandparents to let them discover science via robotics, computer science and artificial intelligence. Approach C [7] was very specific about teaching children (as young as in elementary school) programming with a specific robotics kit. Approach D [8] set secondary school children aged 11–13 into center (with technology in the background) to let them discover their interests and strengths while tackling real-world problems with the help of robotics.

Main-Phase: There were four rounds of questions and a final wrap-up round.

First round—Educational Robotics Landscape 1: Warming up where participants meet and start identifying the landscape (defined in Sect. 2.1) with opportunities and threats.

- What single memorable incident convinced you to teach robotics or why are you teaching robotics to children?
- What is the potential of educational robotics to impact our society?
- What opportunities and threats can be identified?

Second round—Educational Robotics Landscape 2: Participants continue discussing the landscape.

- What are your specific challenges?
- What works really well?
- What depends on the environment?
- What factors influence your concept?

Third round—Educational Robotics Approaches 1: Participants analyse the four presented approaches (see Sect. 2.2).

- Place the concepts A, B, C and D on the ER landscape
- What do method A, B, C or D have in common?
- In which aspects do method A, B, C or D complement each other?

At this point, we offered a categorization (a given set of categories preselected by the authors) to the participants, so they could compare different approaches in a structured manner. This did not work very well. Participants found the categories too high level or generic, or they found more than one category fitting their approach and were not sure which one to choose. Interestingly, they could also not come up with contact points between the approaches themselves, besides age and school level. The same problems occurred in the fourth round but it did not hinder the participants from diving into deep discussions about educational robotics and its problems.

Fourth round—Educational Robotics Approaches 2: Participants continue the analysis by adding other approaches

- Place your concepts on the ER landscape
- What does your method and A, B, C or D have in common?
- In which aspects does your method and A, B, C or D complement each other?

2.3 Analysis

At each of the four world café tables the table host noted findings and summaries of the discussion rounds on a poster. Participants were also encouraged to take notes using post-its and paper notepads. The final poster presentations as well as the concluding feedback round with all workshop participants were audio recorded. Following the workshop all written documents (posters, post-its, notes) were collected and audio recordings were transcribed. The data was analyzed qualitatively [9] by structuring outcomes of the workshop in various rounds. Four categories emerged from this analysis (see Sect. 3).

3 Discussion

3.1 Robots

Robots are powerful tools to motivate children synthesizing knowledge, making it tangible and meaningful. For instance when children code a robot, there is an immediate physical result of the abstract task: the robot performs along the children's instructions. Additionally, teaching subjects are not limited to coding; with robots children can learn about almost anything by trying out, making mistakes, exploring and experiencing. However, the term educational robotics comprises robotic kits as well as robots themselves. In some cases, robots have the role of a teacher or tutor, in other cases, the robot or robotics kit is the tool to teach a concept or subject. It is important to distinguish between these roles and areas of application, as well as the complexity. The discussion among experts reveals that robotics kits and black box approaches should be provided to younger children, older ages can handle robotics platforms with more mechatronics ("white-box") and dive deeper into detail.

3.2 Teachers and Schools

Teachers are the key for the deployment of educational robotics in classrooms. Their involvement and attitude are critical; without them, educational robotics cannot be introduced in classrooms. Some teachers think that children are not able to learn about science and technology, especially on the kindergarten and elementary level, others are afraid of technology, considering it too complex. Teachers are overwhelmed with the vast materials offered, at the same time they lack proper training. In this matter, the expertise of educational robotics experts and researchers is needed. Finally, environmental factors influence the spreading of educational robotics in schools, e.g. money, resources, availability of materials, time constraints and group size play a role. Robots serve different goals and are applied in varied ways according to the school level. In pre-school and elementary, children use a lot of imagination, thus robots have to be contextualized (e.g. using story telling). In junior high school, students use robots as tools to learn different concepts; in senior high school, the focus is building (mechanics and electronics) and coding "real" robots. The university level continues with a more detailed and specialized understanding of the basic robot building concepts. No matter which level, "learning by doing" should be a central aspect. This can be enhanced by specifically designed curricula for all school levels. Given the knowledge and cognitive capabilities of the students, curricula are more needed at the elementary and junior high level to perform classroom activities or workshops with educational robotics. For older ages the "learning by doing" should be concentrated on a specific problem or task. Older students can also be motivated by competitions and after-school activities. However, there should be complementary motivating options, since competitions can also be demotivating also or not appeal to every student.

3.3 Society, Politics and Media

Society, politics and media influence the use of educational robotics as a learning tool. Robots are neutral; experts need to design activities in such a way that children are expressing their thoughts, their ideas, their knowledge and their interests gender independently. Children's expectations are formed by society and media. It then can be frustrating to work on a simple prototype that "only" drives around. This can be countered by lab tours in universities where children see robots in demonstration to understand what they can accomplish when they choose this field and stay on the subject.

3.4 Method "Educational Robotics Café"

The Educational Robotics Café was a novel and fruitful way to hear and discuss in an open atmosphere different approaches, opinions, experiences and perspectives among the workshop participants from different countries. The experts rated the method as helpful and the discussions as very interesting. They were amazed to find out that, in principle, approaches from other colleagues focus on similar questions and problems, although results and the way of doing things differ. They all agreed that strong motivators and proofs were needed to convince important stakeholders like teachers, school administrators, policy-makers, etc. The participants underlined the importance of "making a case" together, and to foster cooperation among experts and educators in educational robotics to transfer this knowledge to the stakeholders. However, the difficulty presented itself in finding common elements that made sharing or comparing possible. The approaches presented in the pre-phase were not described in that way. It would have helped, had they been broken down into

Table 1 Overview of exemplary tags

Overall concept	Strategy	Setting	Target group	Structure
	#top-down	#classroom	#kindergarten	#exploration
	#bottom-up	#after-school	#elementary #K12	#activity
Educational goals	Focus	Teaching goals	Theory/Method	
	#STEM	#robotics	#inquiry-based	
	#literature	#collaboration skills	#project-based	
Materials	Principle	Software	Coding concepts	
	#white-box	#text-based	#high-level	
	#black-box	#visual interface	#low-level	
Evaluation	General	Methodology	Focus	Target group
	#evaluated	#quantitative	#technical skills	#student
	#not evaluated	#qualitative	#interests	#teacher

elements that many approaches or activities have in common. Finally, as authors, we would like to underline the importance of the questions in the method Sect. 2.2. They influence the direction of the conversations and the overall output. Also, we believe that the SWOT analysis did not work well because participants could not find common points of the approaches to compare them. Then, we were amazed to see what started as formal talks of scholars defending their own concepts to become an honest discussion and sharing of experiences.

4 Conclusions

We introduced the “Educational Robotics Café”, a workshop format where experts engage in an open discussion on the educational robotics landscape. One major finding the experts from different domains and countries realized is that, in principle, all have similar goals and experiences although results and the way of doing things differ. As it unfolds, the educational robotics landscape becomes abundant, covering different pedagogical methods, settings, age groups, tools, and activities. The workshop showed that there is a great need to discuss and share experience, yet known categorizations to compare approaches may not break differences down into the necessary micro level. The difficulty rather presents itself in finding common ground or common elements that make sharing or comparing possible. We close our paper with an outlook on how a comparison of different approaches can be made possible by a “tagging” approach to enable better communication between experts from different domains like education or robotics.

5 Outlook: Tagging Approach

Tags can cover key aspects on a micro level and have the advantage of serving as attributes, thus more than one tag can be assigned to one approach and new tags can be added if none seems to fit properly. Table 1 shows exemplary tags divided into four groups. We are aware that this selection does not cover all key aspects in educational robotics. Rather, we intend to stimulate discussion and foster a vital knowledge exchange between researchers and practitioners in the field of educational robotics worldwide.

Acknowledgments Thanks to the participants of the workshop at the RIE 2015 for their contribution. This research has partly received funding from the FWF Science Communication Project WKP42, “Schraege Roboter” (“crazy robots”) as well as from Land Steiermark (“Wissenschaft und Forschung”).

References

1. IEEE Trans. Educ. **56**(1) (2013). Special Issue on Robotics in Education
2. Catlin, D., Blamires, M.: Principles of Educational Robotic Application (ERA). In: Constructionism Conference, Paris (2010)
3. Brown, J.: The world cafe: living knowledge through conversations that matter. PhD thesis. The Fielding Institute (2001)
4. Pahl, N., Richter, A.: SWOT Analysis. Methodology and a Practical Approach. GRIN Verlagr, Idea (2007)
5. Cacco, L., Morro, M.: The hedgehog: how to integrate robotics in natural science scenarios. In: Workshop position paper at International Conference on Robotics in Education (RiE 2015), Yverdon-les-Bains, May 2015
6. Kandlhofer, M., Steinbauer, G., Hirschmugl-Gaisch, S., Eck, J.: Children discover science: robotics, informatics and artificial intelligence in kindergarten and school. In: Workshop position paper at International Conference on Robotics in Education (RiE 2015), Yverdon-les-Bains, May 2015
7. Koppensteiner, G., Vittori, L., Miller, P., Goodgame, S.: Teaching programming on the elementary level. In: Workshop position paper at International Conference on Robotics in Education (RiE 2015), Yverdon-les-Bains, May 2015
8. Lammer, L., Vincze, M.: Crazy robots—Introducing children with different backgrounds and interests to robotics. In: Workshop position paper at International Conference on Robotics in Education (RiE 2015), Yverdon-les-Bains, May 2015
9. Diekmann, A.: Empirische Sozialforschung—Grundlagen, Methoden. Rowohlt, Anwendungen (2007)

A Workshop to Promote Arduino-Based Robots as Wide Spectrum Learning Support Tools

Francesca Agatolio and Michele Moro

Abstract This paper discusses the design of a workshop in Educational Robotics based on Arduino offered to fifty different classes in Italian schools, from primary school to high school. We provided different examples with two robotic platforms. We present some observations about the current conception of Educational Robotics among teachers and students and about the related problems in its introduction in schools. The paper includes a description of the proposed activities and a preliminary evaluation of the results.

Keywords Educational robotics · Arduino · Learning support tool · Conception of robotics

1 Introduction

This paper presents a workshop in educational robotics that is taking place this year in Italy starting in February 2016 and ending in May 2016. The workshop is the result of a collaboration between the Department of Information Engineering of the University of Padova and the Italian association Gruppo Pleiadi[1] whose mission is the scientific dissemination. The reasons behind conception and design of this project are:

1. a strong belief in the potential of educational robotics as a learning tool for the improvement of psycho-pedagogical skills and as an innovative support for traditional theoretical lessons [1–4];

---

[1] http://gruppopleiadi.it/.

F. Agatolio · M. Moro (✉)
Dipartimento di Ingegneria dell'Informazione, University of Padova,
Via Gradenigo 6/A-B, Padova 35131, Italy
e-mail: michele.moro@dei.unipd.it

F. Agatolio
e-mail: francesca.agatolio@dei.unipd.it

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_11

2. the awareness that, despite this potential, educational robotics still plays a marginal and sporadic role in schools [1];
3. the possibility to exploit some current favorable conditions: relatively low-cost robots on the market and suitable for educational purposes; the previous experience of the involved researchers; a relevant demand of teaching/learning innovation coming from the school system and families in Italy.

Nonetheless some limitations still remain and we decided to focus especially on the two of them that we considered the most relevant:

- the lack of resources to buy the right quantity of robotic kits for the entire classes [5];
- the reticence of many teachers, often due to the conception that robotics activities are difficult for students but also too complex for their personal competences [6, 7].

Keeping these points in mind, we started the project Officina Robotica[2] willing and hoping to spread a new, different conception of educational robotics.

The workshop that we will describe in this paper is only the first step of the project and consists in two lessons of two hours each, dedicated especially to automation aspects; the planned recipients of the lessons are fifty classes of different grades and ages, with an average of 25 students for each class.

The main purpose of these lessons is to engage both students and teachers increasing their curiosity about robotics, and to convey the positive conception of robots as a powerful and amusing learning tool. In particular, we want to convince teachers about these facts:

- robotics can be addressed with different levels of difficulty, from simple to very complex;
- the presence of an artifact during the learning process increases the student engagement and encourages active learning (a concept consistent with constructivism and constructionism theories [3, 8]);
- during robotic activities a teacher has a different role compared to traditional lessons: he/she is personally stimulated to find the solution/solutions collaborating with the students and not imposing ready-made solutions;
- it is possible to have an affordable but fully operational laboratory with robots, adaptable both to a computer-based lab and to the usual classroom;
- using robotics, it is very simple and natural to develop links with curriculum subjects (educational robotics does not mean to study robots in technical details).

Regarding the students, the goals of the project are:

- to convey the concepts of sensor, actuator and microcontroller, and how they interact with each other;
- to make them glimpse some practical applications of the theoretical knowledge;

---

[2] http://www.officinarobotica.it/.

- to inform the students of the existence of affordable robotic tools that can be further explored out of school.

As better explained below, for the purposes of this workshop many reasons convinced us to adopt Arduino-based robots. In spite of our previous experiences, the main factor not to use the more common LEGO Mindstorms robot for the workshop was the limited allocated time and the cost.

2 Structure of the Workshop

The workshop activity originates from the project Officina Robotica and is being developed thanks to a collaboration between the Department of Information Engineering of UNIPD and Gruppo Pleiadi.

The purpose of the activities is to introduce among teachers and students a new and stronger perception of the wide possibilities educational robotics can convey in school along with the idea that robotics is a tool accessible and ideal for all ages; the project is therefore addressed to all school levels, from primary school to high school.

Keeping this objective in mind, we presented the workshop to fifty classes among thirty different schools, located in different cities of north (78 %) and central (22 %) Italy, namely:

- twenty primary school classes (1st–5th grades);
- twenty junior high school classes (6–8th grades);
- ten high school classes (9–12th grades).

The project involved among 1250 students and 40 teachers. We selected the schools according to the order in which they have requested to participate. They were both public schools and private schools. Regarding high schools, almost all the teachers who asked to participate belonged to technical schools. The age of the students was highly variable, especially for primary school (40 % of the students was 6–8 years old). Regarding junior high school and high school, among 90 % of the students were 12–15 years old. Almost always the students who took part in the workshop were not voluntary but they belonged to a class chosen by the teacher.

The workshop is organized into two lessons of two hours each, with a break of a month between the first and the second. The first lessons took place from the second half of February 2016 to the second half of March 2016. The second lessons were scheduled in April 2016. For almost all the students who take part to these activities, this is the first experience with robotics and also mostly for programming.

3 The Instrument

3.1 *The Tool: Why Arduino*

Arduino-based or compliant robots were our choice due to some important positive aspects:

- relative easiness of construction;
- low-cost components, giving the students the possibility to replicate experiences at home;
- a variety of usable programming environment (beyond the standard Arduino IDE,[3] Ardublock,[4] Scratch for Arduino,[5] Snap for Arduino,[6] Mblock[7]).

Arduino is specifically linked to the Maker philosophy: it is in fact especially suited to the improvement of manual skills and active experimentation. The goal is to make students reflect on the strong relation between the theoretical contents they just learned and a direct real experience, in order to lead them to think about the world in an informed and scientific way, as opposed to a "magical one" [9]. The rich set of sensors and actuators compatible with Arduino is well appropriate to this purpose. The cost-effective quality of Arduino was particularly appreciated in our case in order to provide more kits during the lessons, one for every two/three students. It is also a good premise to convince schools and single students to invest in robotics and thus to promote its wider diffusion. Our experience has already showed also its flexibility: Arduino is easily adaptable to different levels of competence and school stages; it can be used both to introduce the fundamentals of robotics/electronics and to develop more complex projects.

Concerning junior high schools and high schools, we opted for the usual Arduino hardware (Arduino UNO, with a starter kit included); the Arduino electronic board has to be assembled on a mobile platform, a choice we made with the hope to increase the students involvement by showing them a structure closer to what we observed their expectations about a robot were.

For the application of our workshop in primary schools we chose Mbot 2.4G version, an Arduino-compatible platform which presents some further interesting characteristic:

- the kit includes a set of sensors and actuators, some on board, some as additional extensions on mini-cards that makes the interaction between the robot hardware and the ambient clear, immediate, and partly referable to interactions common in human beings;

---

[3] https://www.arduino.cc/.

[4] http://blog.ardublock.com/.

[5] http://s4a.cat/.

[6] http://s4a.cat/snap/.

[7] http://www.mblock.cc/.

- among the other solutions, it can be controlled through a customized version of Mblock, a graphical environment based on Scratch.

Moreover it can be operated using an infrared remote controller and this makes also suitable for little children (aged 6–7).

3.2 Programming Languages

One of the main issues in the realization of the workshop was the choice of the programming language and how much time to allocate to the programming part. In Italian schools basics of programming are taught only in some high schools, and we are conscious that it is very difficult to introduce programming in just two lessons; on the other hand, we firmly believe that robotics without any programming tasting loses its deep meaning, so we decided to introduce a dedicated section in order to show the fundamental role of programming in the robots' behavior.

Eventually we chose the following block-based programming environments for the different school levels:

- Primary school: Mblock (that it is based on Scratch)
- Junior high school: Scratch for Arduino (S4A)
- High school: ArduBlock

Considering that most students do not have any previous knowledge about programming, we chose to mediate heavily this part.

In junior high school and in high school we use an introductory activity (the smart lamp) to give a direct and simple demonstration. Students are first asked to replicate, on their pc, a project with an already tested piece of code for a first familiarization. In the following activities we gave them driving sheet with a support for the programming.

In primary school we make the choice to avoid so young students were challenged by programming technicalities in order to focus on the robot behavior and on command rich semantics. Therefore the experience is first devoted to design robot actions in terms of simple sequences of commands: this phase is supported by some worksheets illustrating in simple form the library of selected commands. These sequences are preliminarily tested using a body syntonic approach (the teacher or one classmate "executes" commands like a robot). Then the teacher codes the sequence using the graphical environment and the students observe if the robot precisely reproduces what is expected.

4 Activities

The collaboration with Vivigas&Power,[8] which is supporting the Officina Robotica project, asked to orientate the activities towards the energy theme. We considered Arduino well adaptable to this theme.

The competences that we wanted to develop through the workshop are:

- understanding the structure of a robot and the relation between sensors, actuators and the microcontroller;
- understanding the role of the human pilot in controlling the robot through the programming language;
- understanding the fundamentals of building electronic circuits.

For the realization of the workshop we had to keep in mind that:

- many of the students do not have any knowledge in Arduino and robotics in general;
- the selected classes presented a highly variability in age and probably also in level.

Therefore, for each of the three school levels, we chose to design a large set of activities different in complexity (both with regards the automation aspect and the programming aspect). In this way, we can easily calibrate any lesson depending on the level and the response of the different classes.

Anyway the first activities have the aim to make the students feel comfortable using robots, so these activities are focused on the manual building of very simple electronic circuits and do not involve any actual programming of a robot.

4.1 Primary School Activity Examples: “Macchina Scribacchina” and an Example of Activity with MBot

For primary school, the first part of the workshop is strongly dedicated to improve manual skills. For this we chose to start assembling some simple circuits using LEDs, DC motors and battery. Then children could use this knowledge to build the artifact “Macchina Scribacchina” (literally Scribbling Machine) [10] (Fig. 1), a moving machine obtained by assembling a plastic glass with a DC motor and four markers. The vibration of the motor is transferred to the glass and to the markers so that they leave some circular traces on the path.

After this activity, we introduced the robot MBot. The first activity was to analyze the behavior of “Nocturnal MBot”. In this case, students didn’t have to program the robot but only to observe the result of the program. “Nocturnal Mbot” moves only if the light intensity goes down a certain threshold. So the first impression is of a magical artifact that dances and blinks only when it is dark and we usually cannot see it [11]. In this way we want to introduce children to robotics starting from a conception of a robot close to a toy. After this first impact we go to unravel the

---

[8] http://www.vivigas.it/.

Fig. 1 “Macchina scribacchina”

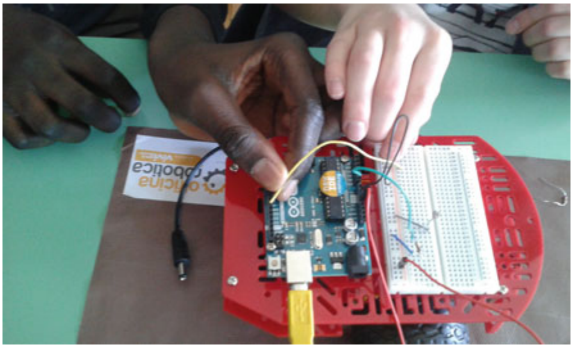


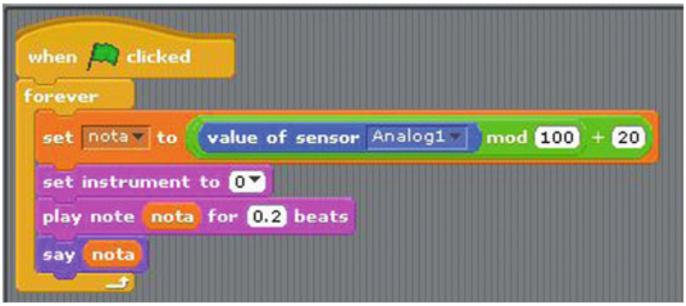
Fig. 2 Smart lamp

reasons behind the robot behavior introducing a scientific explanation. The children are introduced to the sensor and actuator concepts and their role in designing a robot control program.

4.2 Junior High School Activity Example: The Smart Lamp

This is the first activity actually introducing to programming language. Keeping in mind the saving energy theme, we create a lamp that lights up only if the light intensity is low enough for the environment to be considered darkness (Fig. 2).

To realize this object, a simple LED and a photoresistor are sufficient; the aim of the activity is to introduce the concept of analog input and the necessity to understand sensor’s values and how to translate electric signal into something which is understandable to humans. The students meet for the first time some typical structures of a programming language, i.e. creation and use of variables, the if/else command, etc. The students are also encouraged to establish autonomously the threshold that allows them to distinguish darkness from light. Moreover, they are induced to reflect on the difference between the last activity, where the LED was not controlled by anyone, and this activity, where the lightning of the LED is programmed. We want



```
when [flag] clicked
forever
    set nota to (value of sensor Analog1 mod 100) + 20
    set instrument to 0
    play note nota for 0.2 beats
    say nota
```

Fig. 3 Playing with the light

to make explicitly clear that the behavior of any robot strongly depends on our given instructions.

In junior high school, after the first demonstrative phase (the students have to copy already working code), in the second part of the activity we want the pc to play a musical "instrument" using the variations of the light sensor mounted on the robot and using a little of mathematics (Fig. 3). In this activity we use the same circuit of the "smart lamp" and students can concentrate only on the programming part. For this exercise we give them a worksheet for the programming part, so they can reflect on it. The reaction to the outcome is very enthusiastic: the students are rewarded and they can see the potentialities of the robot. In particular, the introduction of sound elements is very engaging for students who usually present serious learning problems, giving them a chance of actively participate (e.g. choosing the musical instrument, proposing tunes).

4.3 High School Activity Example: Ecological Wake up and Sunflower Robot

In high school, after the introduction to programming through the "smart lamp" activity (see Sect. 4.2), we proposed the "ecological wake up" activity. The students used a photoresistor and a buzzer to create an alarm clock that starts playing when there is a sufficient quantity of light. We provided them a driving sheet for the programming and they had to complete autonomously the code. We noticed that most of them found hard to distinguish between the setup phase and the loop phase (Fig. 4).

The activity "sunflower robot" will be carried out in the second part of the workshop. We create with Arduino a robot that follows a light source. This project is strongly related to the eco-save question and can be used to create a solar panel that, during the day, moves following the sun. To implement the project, we use two photoresistor oriented so that they form an angle of 90°. When the light source is closer to one of the two sensors, the robot turns in order to stay in line with the source

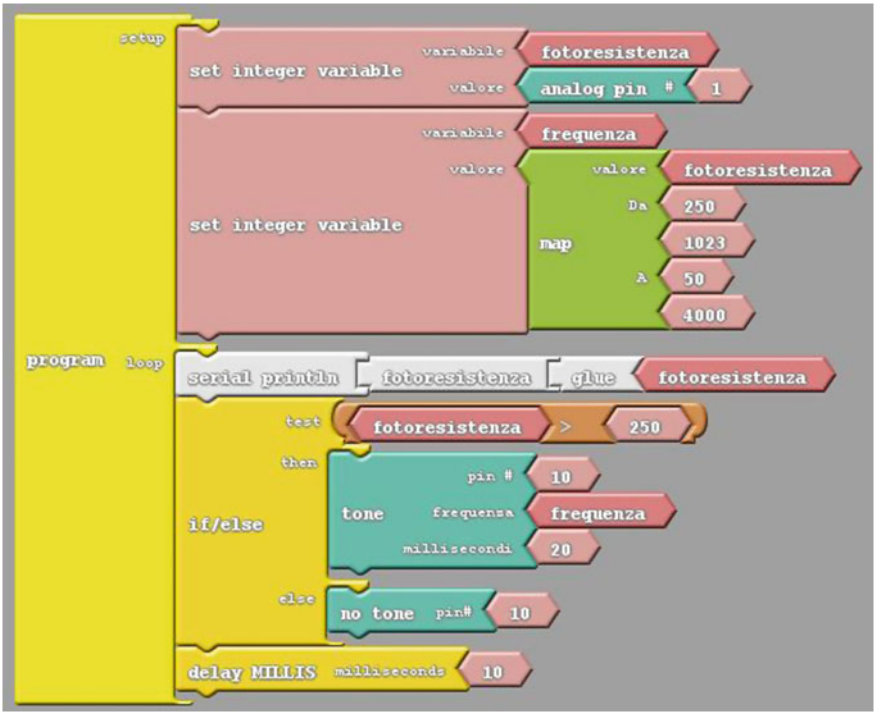


Fig. 4 An example of common error in programming

[12]. This project is strictly connected to practical needs and allows an immediate understanding about the possibilities when using robotics and theoretical knowledge in a daily context. This can be useful to improve the engagement of students and to justify the effort of the activity.

5 Evaluation of the Workshop: Preliminary Considerations

5.1 The Evaluation Tool

The evaluation of the workshop will be done at the end of the second lesson, through a short questionnaire that will be issued to the involved teachers. The questionnaire aims at evaluating essentially three points:

- the conception about the potential of educational robotics as a learning support tool;
- the intention to introduce an educational robotics curriculum;
- the level of satisfaction about the workshop.

The respondents are asked to answer both to closed questions (yes/not) and open questions ("Do you think that educational robotics can be a useful support learning tool? Why?", "Do you think about introducing some robotics activities in your lessons? Why?").

Regarding the students, we considered meaningless to evaluate any improvement in learning after only two lessons. Therefore, we preferred to give them the opportunity to elaborate and to reflect about the recent knowledge through a contest, that is to design a robot with the purpose to help the ambient, using strongly the concepts of sensors, actuators and energy in general. In junior high school and high school, students can present an original project using an hardware like, for example, Arduino and LEGO Mindstorms, or a report of an activity of the workshop. Instead, in primary school, students can simply describe and draw the imagined robot. A contest is not an evaluation tool but, using a contest, we would like:

- to have a general feedback about the understanding of the conveyed concepts;
- to induce students and teachers to carry on a part of the workshop autonomously, without the external intervention.

To encourage the schools participations, we reserved a prize for the best project (a kit of five Arduino for secondary schools and a kit of five MBot for primary schools).

5.2 The Role of the (Curricular) Teacher

One of the greatest problems we met introducing robotics during this workshop was the rapid demoralization of students when they face a failure (especially at junior high school and high school). This is probably due to the fact that many of the arguments are introduced ex novo, but it could also be closely related to the different learning approach of robotics compared to traditional lessons. In particular, at the beginning students show a strong resistance to the trial-and-error approach that is underlying to educational robotics: they seem to be afraid of the possibility to search for a solution by themselves (they often ask the teacher for support even before making any attempt to solve the task) and most of them don't seem interested in spontaneously exploring new solutions, unless the teacher compels them to do it. This is particularly evident in female students who sometimes appear less engaged in the subject.

In view of the above, we consider the role adopted by the curricular teacher fundamental for the positive involvement of the students [13, 14]. The teacher in such a constructivist learning approach like the one described above does not work as a teacher—authority that transfers ready knowledge to students—but rather acts as an organizer, coordinator and facilitator of learning for students. He/she gives the guidelines of the activity, provides to students with many materials for thoughts and observes their learning process. His/her presence should be very discrete and his help should be offered only when necessary. He/she should allow students to work with creativity, imagination and independence [15]. Acting like this, the teacher should:

- encourage the students to adopt an active role in the learning process, based on direct experience without fear of making errors;
- help to build their self-confidence, trying to make them feel that his/her help is less crucial and the stepwise refinement of solutions through a trial-and-error process is the main guideline.

Finally, in order to increase their involvement, the teacher has to mediate the meaning of the experience underlining both the links with reality and the connection with the theoretical knowledge learned at school; in this way he/she realizes a mutual contribution amongst robotics and theoretical lessons aimed at increasing the students' interest.

5.3 Preliminary Results

Until now about ten classes have completed both the two lessons, but we can anyway make some preliminary observations about the achieved goals.

In general, our perception of the project's outcome is positive. Due to the diversity of the involved schools and classes (grade, number of students in each class, participation of the teacher), we can observe a different feedback regarding the robotics conception that we try to convey. Of course in classes with a large number of students (more than 25) it can be hard to take on such a workshop: numerous requests of help and support can concurrently come from dfferent groups and this would suggest the presence of more tutors. The risk of that is to observe a certain degree of discouragement. Even if it is not surprising, we registered the more enthusiastic feedback in primary school (especially in students), while junior high school confirmed as belonging to the most critical range of ages (in particular it resulted more difficult to design activities with the right level of complexity using Arduino).

Regarding students, in general we noticed a sincere enthusiasm, despite some difficulties (see Sect. 5.2); moreover some of them seemed to be interested in the possibility to carry on autonomously the activity out of the school. Regarding teachers, we perceived an high appreciation as well. This confirmed by the partial data obtained by the questionnaire. On the basis of the opened questions and other general observations, it results that the good level of satisfaction is due especially to:

- the engagement of their students in using robots;
- the awareness that educational robotics takes advantage of diverse skills, which don't usually come up in traditional education. In particular, they pointed out that in many cases, among the students that better succeeded in the activity, there were surprisingly students with an otherwise low school performance.

However, despite this positive reaction, several teachers revealed they are not intentioned to introduce educational robotics in the curriculum because they are not confident of their competences (Fig. 5). Also among the teachers that said yes to this question, most of them declared to not feel able to do it without an external support.

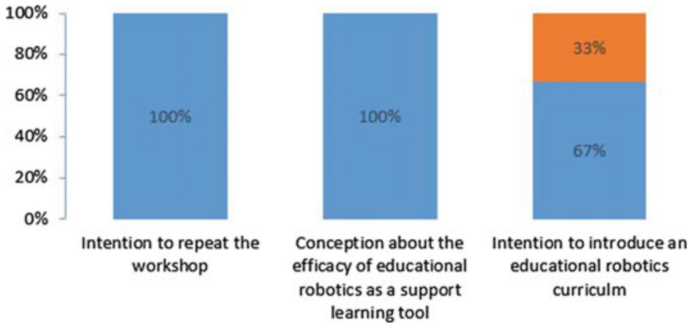


Fig. 5 Partial results from the questionnaire

6 Conclusions

At the moment of writing, all the fifty selected classes have already taken part in the first meeting of the workshop and about ten classes have taken part also in the second meeting. All the schools involved in the workshop asked voluntarily to participate, often due to a specific interest of teachers qualified in scientific subjects. In general, our perception of the project's outcome is positive: after an initial resistance, most students felt rewarded by the results and seemed impatient to continue the experience with a next lesson. The feedback of the involved teachers was positive, as well, and, in particular, their conception about the potential of the educational robotics as learning support tool seemed to be increased. Despite this, most of them not yet feel able to carry out autonomously educational robotics activities.

A more complete evaluation of the workshop will be done after the end.

Acknowledgments We thank the power and gas society Vivigas&Power, that supports the Officina Robotica project.

This work was also partly supported by the project: ERASM: Educational robotics as a validated mindtool: methodology, platforms, and an experimental protocol, code: CPDA145094, funded by the University of Padova.

References

1. Alimisis, D.: Educational robotics: open questions and new challenges. Themes Sci. Technol. Educ. **6**(1), 63–71 (2013)
2. Sullivan, F.: Robotics and science literacy: thinking skills, science process skills and systems understanding. J. Res. Sci. Teach. **45**(3), 373–394 (2008)
3. Mikropoulos, T., Bellou, I.: Educational robotics as mindtools. Themes Sci. Technol. Educ. **6**(1), 5–14 (2013)
4. Eguchi, A.: Educational robotics for promoting 21st century skills. J. Autom., Mob. Robot. Intell. Syst. **8**(1) (2014)

5. Avvisati, F., Hennessy, S., Kozma, R.B., Vincent-Lancrin, S.: Review of the Italian strategy for digital schools. OECD Education, Working Papers 90, OECD Publishing (2013)
6. Tondeur, J., Braak, J., Sang, G., Voogt, J., Fisser, P., Ottenbreit-Leftwich, A.: Preparing pre-service teachers to integrate technology in education: a synthesis of qualitative evidence. Comput. Educ. **59**, 134–144 (2012)
7. Di Battista, S., Menegatti, E., Moro, M., Pivetti, M.: Introducing educational robotics through a short lab in the training of future support teachers. In: 6th International Conference on Robotics in Education, RiE 2015, Yverdon-les-Bains, Switzerland (2015)
8. Alimisis, D., Arlegui, J., Fava, N., Frangou, S., Ionita, S., Menegatti, E., Monfalcon, S., Moro, M., Papanikolaou, K., Pina, A.: Introducing robotics to teachers and schools: experiences from the TERECoP project. In: Clayson, J., Kalas, I. (eds.) Proceedings for Constructionism, pp. 16–20. Paris, France (2010)
9. Roy, D., Gerber, G., Magnenat, S., Riedo, F., Chevalier, M., et al.: IniRobot: a pedagogical kit to initiate children to concepts of robotics and computer science. In: 6th International Conference on Robotics in Education, RiE 2015, Yverdon-les-Bains, Switzerland (2015)
10. The Tinkering Studio. http://tinkering.exploratorium.edu/
11. Dickel, M., https://www.youtube.com/user/MechDickel
12. Alfieri, M., http://www.mauroalfieri.it/
13. Agyei, D., Voogt, J.D.: Exploring the potential of the will, skill, tool model in Ghana: predicting prospective and practicing teachers use of technology. Comput. Educ. **56**, 91–100 (2011)
14. Ertmer, P., Ottenbreit-Leftwich, A., Sadik, O., Sendurur, E., Sendurur, P.: Teacher beliefs and technology integration practices: a critical relationship. Comput. Educ. **59**, 423–435 (2012)
15. Alimisis, D.: Exploring paths to integrate robotics in science and technology education: from teacher training courses to school classes. IJREA **2**(2), 16–23 (2012)

Robotics in School Chemistry Laboratories

Igor M. Verner and Leonid B. Revzin

Abstract This paper proposes an approach to integrated learning of robotics and chemistry implemented in a laboratory course, in which high school students majoring in mechanical engineering are involved in the development of laboratory robotic devices and use them for chemical experiments. We consider learning activities at different stages of the course and characterize students' engagement along the stages.

Keywords Educational robotics · Chemistry · Experiential learning · Laboratory automation · High school

1 Introduction

Modern chemical research and development laboratories widely exploit automatic devices, hence many universities teach students to use automation technology, and some already stopped teaching traditional manual methods [1]. In line with this progress, high schools are making efforts to modernize their science laboratories and transform them into technology enhanced laboratory learning environments [2, 3]. Educators believe that new automation technologies can make laboratory experimentation more efficient and accessible, providing significant educational advantages, especially in facilitating constructivist inquiry-based learning [4] and in fostering students' higher order thinking skills [5].

Currently, many school laboratories are equipped with computerized systems which provide automation of experimental data collection and analysis by means of sensors and data loggers [5]. It should, however, be noted, that while these computerized systems provide automated data management, all manipulation operations required for chemical experiments are still performed manually. Budget constraints

---

I.M. Verner (✉) · L.B. Revzin
Faculty of Education in Science and Technology,
Technion – Israel Institute of Technology, Haifa, Israel
e-mail: ttrigor@technion.ac.il

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_12

limit opportunities for schools to purchase commercial automation systems. The lack of automation of manual operations causes difficulties in organizing laboratory experimentation that answers the safety, accuracy and timing requirements. Therefore, it became necessary to find affordable solutions for the automation of manipulations in school chemistry laboratories.

Our research proposes and investigates an approach in which students experientially learn chemistry, using data management systems and self-constructed computer-controlled devices for automation of basic manual operations. We developed a learning environment which includes data loggers and sensors, robot construction kits, and automation devices constructed by the authors and students. We conducted studies of learning in the proposed environment for two categories of high school students: 11−12th graders studying chemistry at advanced level, and 10−12th graders majoring in mechanical technology. We assert that through practice in construction and application of automation devices students can acquire knowledge and skills in both chemistry and technology.

2 Conceptual Framework

The use of robotic learning environments in school science laboratories has rapidly expanded in recent years. This trend has motivated research in educational robotics and in the broader context of learning in technology enhanced learning environments (TELE). Educational processes in TELE are mainly in the form of constructivist inquiries that foster understanding concepts of science and technology, and the development of research skills [4]. In such environments the students construct knowledge in science and technology through practice in creating and operating technological tools. Researchers point to the potential of TELE to foster interest in learning [6] and present evidence on the increase of students' motivation [7]. Kim et al. [4] note that in technology enhanced learning, student motivation should come together with teacher's scaffolding. Linn [8] points that technology enhanced learning environments can provide guidance to students.

Dori and Kaberman [9], based on their research findings, claim that TELE can be especially effective in supporting learning of low achieving students. According to Girasoli and Hannafin [10], scaffolded practice in TELE promotes the development of student's self-efficacy.

New science curricula emphasize the need for personalized learning adapted to thinking styles and learning strategies of each learner. Educators argue that through the application of advanced technologies learning can be customized even for large classes [11].

Most studies in which learning in school science laboratories is enhanced by computer-controlled devices, consider situations when the devices are purchased from manufacturers. Little research is about courses in which students under teachers' guidance construct such devises and use them for science experimentation.

The Technion Center for Robotics and Digital Technology Education has performed a series of studies of integrated learning of robotics and physics [12, 13], biology [14], and engineering [15]. These studies have been conducted in school and Technion robotics laboratories. In the study presented in this paper we consider a case of learning in school robotized chemistry laboratories. Some robotic devices developed for automation of manual operations in chemistry experiments and preliminary analysis of their educational effectiveness were presented in [16, 17]. In this paper we focus on results of our educational study of students' perception of the automated laboratory environment and engagement in experiential learning.

3 Robotized Laboratory Environment

We enriched the standard school chemistry laboratory by sensors and data loggers, robot construction kits, and devices for automation of basic manual chemical operations. We presented in [16, 17] the first devices constructed by the authors and the ways they were improved, based on students' feedback. Based on this experience we engaged students in construction of new devices, some of which are presented below.

The turntable (Fig. 1a) carries out rotations on given angles, the sensors holder (Fig. 1b) puts the electrode into the solution and puts it up after the

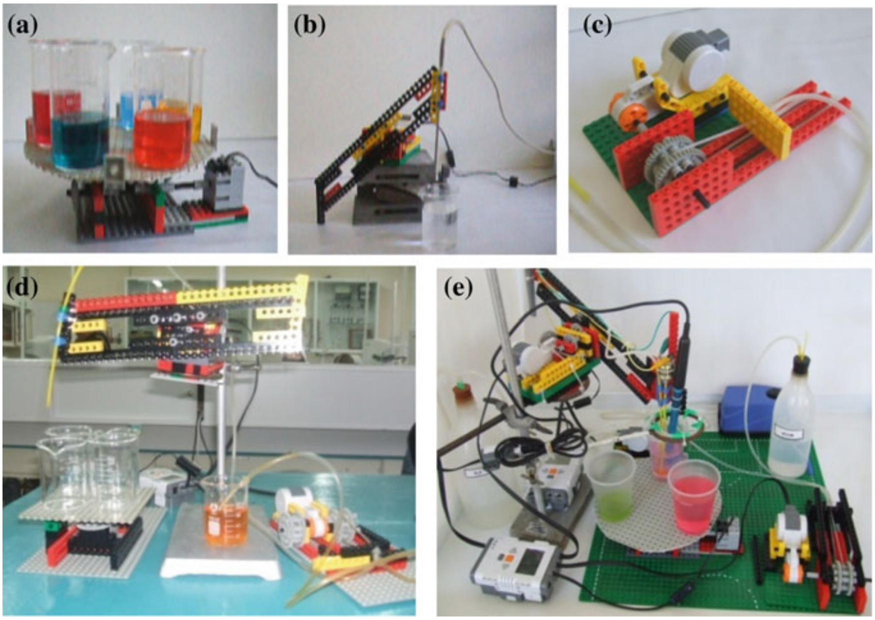


Fig. 1 **a**. Computer controlled turntable; **b**. Computer controlled sensors holder; **c**. Robot-dispenser; **d**. Robot-titrato; **e**. Automatic titration system

experiment, and the peristaltic pump delivers a solution at the given computer controlled rate.

Both computer controlled devices were constructed using LEGO NXT, each of them by two 10th grade students majoring in technology. Figure 1d presents a robot-dispenser for automatic aliquoting of chemical solutions. This robot was constructed by other two tenth graders. The dispenser includes a turntable, a sensors holder, and a peristaltic pump, all controlled by one LEGO NXT controller. Figure 1e shows a robot for simultaneous automatic titration of a number of probes. It was constructed by two 12 graders majoring in chemistry and computer science. In addition to a turntable, a holder, and a pump, it also includes an automatic sensor-washing device, and the Vernie pH sensor. The robot system is controlled by two NXT controllers connected with each other via the Bluetooth communication.

4 The Study

The goal of our educational study was to investigate learning through practice in constructing robotic devices and using them for studying high school chemistry through laboratory inquiry. We were interested to explore our approach for teaching chemistry at both advanced and basic study levels.

Two categories of participants of the research were: students from two comprehensive high schools studying advanced chemistry (172 students of grades 11–12, age 17–18), and students majoring in mechanics technology from a vocational HS (12 male students studied our course in grades 11–12) and from a comprehensive HS (14 male and 11 female students of grade 10, age 16). The students in the former category were high achievers, while in the latter one, were related to the group of youth at risk.

Our results related to the first category of students were presented in [16, 17]. In this paper we will consider learning chemistry by technology students and focus on the following research question: What are specific features of engagement of the technology students in the course and what factors of engagement were dominant at different stages of the course?

To answer the research question, we used questionnaires, observations and interviews to follow up after changes in behavior of the technology students during the course and their perceptions of the automated chemistry laboratory.

5 Learning Process

The learning process in the course given to two groups of technology students consisted of the same eight stages described below.

Stage 1: Frontal teaching. During the first three weeks we tried this method, as it is conventionally used in chemistry education. The method was found unsuitable for this category of learners, characterized by limited background, learning and communication skills, and low motivation for learning science.

Stage 2: Practice in construction of automation devices. We put the study of chemistry concepts aside and offered the students a constructive assignment: to develop a LEGO-based automation device. Each student chose a device from the given list and constructed it under the teacher's guidance (Fig. 2a). Through this practice the students learned about gears, mechanisms, and motors.

Stage 3: Device calibration experiments. The teacher posed to the students questions on practical usability and accuracy of the constructed devices, and he suggested to answer the questions experientially by calibrating the developed devices. He taught the students to use Fourier sensors and the data logger, as well as laboratory devices, in order to do calibration experiments in the automated mode. Here, for the first time, the students made a scientific experiment. On the need-to-know basis they got familiar with physics concepts (velocity, force, pressure, temperature, mass, volume, density) and chemistry concepts (atom, molecule, state of matter, gaze laws). They learned to analyze experimental data by means of Excel.

For example two students made an experiment aimed to calibrate the computer controlled peristaltic pump that they developed. They measured the mass of water automatically pumped during certain time intervals, using the OhausB100 digital balances (Fig. 2b). From the data analysis the students found that the solution feed-rate for the given motor power is constant (see Fig. 3a). They also determined the feed-rate dependency on the motor power, shown in Fig. 3b. During these experiments the students had to help each other and this way developed partnerships that continued at the next stages.

Stage 4: Industrial tour. Each of the two student groups visited a chemical plant and then participated in the discussion of the observed technological processes and the underlying chemical phenomena. The teacher demonstrated simple experiments related to the phenomena and initiated a discussion in which the students developed ideas on how their devices can be used to perform the experiments in the automated mode. At this stage the students acquired concepts of chemical reaction and its equation, process control, and heat exchanger.

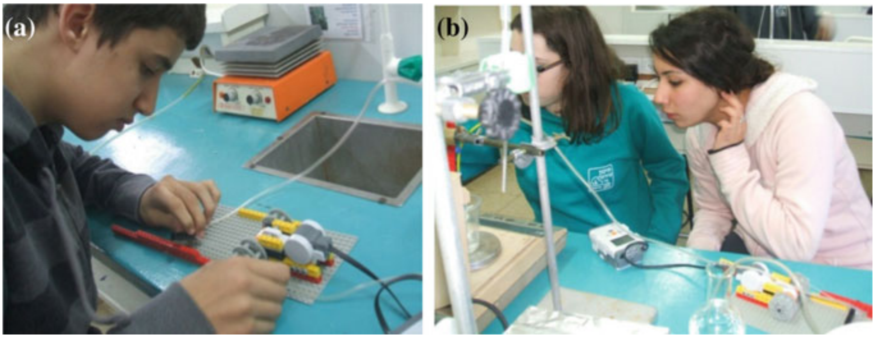


Fig. 2 Peristaltic pump models: **a**. Construction; **b**. Calibration experiment

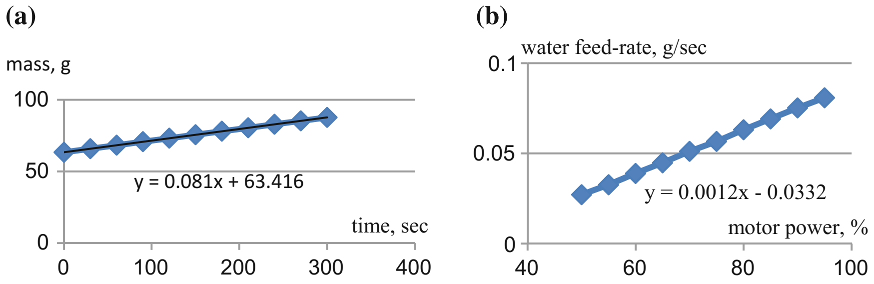


Fig. 3 Solution feed-rate: **a**. Determination; **b**. Dependency on the motor power

Stage 5: Concept development and experimental design. The teacher demonstrated a series of chemical processes and posed the project assignment: to implement them using the automation devices developed by the students. The students performed the projects in the groups of two. Each group was guided to formulate the inquiry question, design an experiment, and specified needed materials.

Stage 6: Creation of an automated experiment setup. The students presented in class the devices that they developed at the previous stages and discussed using them in their projects. Based on these devices, each group designed, constructed, and calibrated a modular robotic system for the assigned chemical process. The implemented processes included titration, sedimentation, and neutralization of dangerous materials.

Stage 7: Chemical inquiry. Each group of the students applied the developed system to make a series of experiments in which they collected and analyzed data. At this stage the students learned to program LEGO controllers, process data by means of the MultiLab software. The students practically used mathematical concepts of linear function, graph, slope, as well as chemistry concepts, such as saturated and diluted solutions, acid-base indicator, chemical reaction speed, and pH.

Stage 8: Project report and presentation. The students analyzed results of the experiments, and summarized the findings in the project reports. At this stage they conceptualized the studied science and technology concepts. In order to get credit for the project in the school matriculation certificate the students had to present it for external examination of the Ministry of Education. The two groups of students presented their reports as school graduation projects in mechatronics. One of the groups also passed the matriculation exam in basic chemistry. Majority of the students successfully passed the exams.

6 Learning Engagement

Students' engagement is an important characteristic of learning and a predictor of its outcomes. It characterizes student's attention, interest and effort in the work of learning [18]. Engagement is expressed through learning behavior, emotional

Table 1 Structures of learning engagement [18, 19]

Engagement structure	Description
Get The Job Done	The student's motivating desire is to satisfy a sense of obligation to complete an assigned task, to correctly follow the given instructions
Look How Smart I Am	The motivating desire is to impress classmates and achieve positive self-regard by demonstrating the ability, knowledge, and intelligence
Check This Out	The motivating desire is to get a payoff or benefit
I'm Really Into This	Here the desire emerges when the student is excited by the learning activity
Don't Disrespect Me	The desire emerges when meeting a perceived threat to the student's dignity, status, or self-respect
Let Me Teach You	The student desires to help to a classmate who faces difficulties in performing learning tasks
Pseudo-Engagement	The desire is to look good to the teacher or to peers by seeming to be engaged

involvement, and intellectual effort. It can change during learning practice and depends on its outcomes. Goldin et al. [18] proposed a theory of engagement structures for systematic observation of typical student behaviors in class. They presented a list of 12 engagement structures revealed through observation in mathematics lessons. Verner [19] proposed to use the theory of engagement structures in the studies of experiential learning in robotic environments. In this paper we use these structures to follow-up the progress of student engagement in our course. Among the twelve structures proposed by Goldin, seven were observed in our case study. They are listed and described based on [18, 19] in Table 1.

From observations throughout the course and by applying the inductive method [19] we found three additional new engagement structures:

I'll Do Something Else The student's motivating desire is to substitute a given assignment by a personally meaningful one, without compromising the difficulty.

Don't Want To Learn It Students' reluctance to perform learning assignments imposed against their will.

It's Interesting To Discuss The students find the learning experience interesting and desire to discuss it.

In order to follow-up the progress of learning engagement along the learning process stages discussed in Sect. 5, we analyzed indications of presence of the ten engagement structures. Our results are presented in Table 2.

The table indicates a significant behavioral change in the students. At the first stage, when the course was delivered through conventional frontal teaching, the students actually disengaged from the learning and even resisted it. At the second and third stages, when the teaching method changed, the students had become engaged in construction and calibration of automation devices and got a desire "to do the job well" and even implement their own ideas. The industrial tour (Stage 4) prompted the students to discuss with peers the seen chemical processes and this

Table 2 Progress of learning engagement during the course

Stage Engagement structures	1. Frontal teaching	2. Practice in construction of automation devices	3.Device calibration Experiments	4. Industrial tour	5.Concept development and experimental design	6. Creation ofan automated experiment setup	7. Chemical inquiry	8. Project report and presentation
Pseudo-Engagement	🟨							
Don't Disrespect Me	🟨							
Don't Want to Learn It	🟨							
Get The Job Done		🟨	🟨					🟨
I'll Do Something Else		🟨						
Check This Out		🟨	🟨			🟨		
It's Interesting To Discuss				🟨				
I'm Really Into This					🟨	🟨	🟨	
Look How Smart I Am						🟨		
Let Me Teach You						🟨		

way contributed to building a learning community. The chemistry inquiry activities at Stages 5–7 deeply engaged and excited the students. We note the multifaceted impact of creation of the automated experiment setups (Stage 6) that prompted students' to demonstrate their mastery and help peers. At the final stage of the course (Stage 8) the students showed persistence in preparing reports and presentations satisfying the criteria of the Ministry of Education. Results of the external examination of the students' projects support the notable presence of the engagement structure "Get The Job Done". 80 % of the school students from the first group and 91 % from the second group successfully completed their graduation projects in mechatronics. 40 % of the first group students by their initiative took the matriculation exam in basic chemistry and successfully passed it.

7 Conclusion

We developed, implemented, and evaluated an approach to integrated learning of science and technology, in which high school students perform chemical experiments in a laboratory enhanced by self-made robotic devices for automation of manual operations. Our experience shows the considerable potential of this approach for students studying advanced chemistry (presented in [16, 17]), and technology students studying a basic chemistry course. The reported case study indicated that for technology students, to whom the conventional frontal teaching was ineffective, the proposed approach, based on the development of robotic devices and using them for chemical experiments, was engaging and contributed to significant outcomes in learning chemistry. It should be noted, that in the recent survey of chemistry education in technology enhanced laboratory environments our work was the only one that studied the use of robotics for automation of manual operations [20]. Based on positive results of the study, we have started experiments of applying the proposed approach to teaching basic physics in a laboratory automated environment to 10th graders (age 16) majoring in mechanics.

References

1. Sterling, J.D.: Laboratory automation curriculum at Keck Graduate Institute. J. Assoc. Lab. Autom. **9**(5), 331–335 (2004)
2. Wang, F., Hannafin, M.J.: Design-based research and technology-enhanced learning environments. Educ. Technol. Res. Dev. **53**(4), 5–23 (2005)
3. Wu, H.K., Huang, Y.L.: Ninth-grade student engagement in teacher-centered and student-centered technology-enhanced learning environments. Sci. Educ. **91**(5), 727–749 (2007)
4. Kim, M.C., Hannafin, M.J., Bryan, L.A.: Technology-enhanced inquiry tools in science education: An emerging pedagogical framework for classroom practice. Sci. Educ. **91**, 1010–1030 (2007)
5. Barnea, N., Dori, Y.J., Hofstein, A.: Development and implementation of inquiry-based and computerized-based laboratories: reforming high school chemistry in Israel. Chem. Educ. Res. Pract. **11**, 218–228 (2010)
6. Ainley, M., Hidi, S., Berndorff, D.: Interest, learning, and the psychological processes that mediate their relationship. J. Educ. Psychol. **94**, 545–561 (2002)
7. Mistler-Jackson, M., Songer, N.B.: Student motivation and Internet technology: Are students empowered to learn science? J. Res. Sci. Teach. **37**(5), 459–479 (2000)
8. Linn, M.: Technology and science education: starting points, research programs, and trends. Int. J. Sci. Educ. **25**(6), 727–758 (2003)
9. Dori, Y.J., Kaberman, Z.: Assessing high school chemistry students' modeling sub-skills in a computerized molecular modeling learning environment. Instr. Sci. Int. J. Learn. Sci. **40**(1), 69–91 (2012)
10. Girasoli, A.J., Hannafin, R.D.: Using asynchronous AV communication tools to increase academic self-efficacy. Comput. Educ. **51**(4), 1676–1682 (2008)
11. Corcoran, T., Silander, M.: Instruction in high schools: the evidence and the challenge. Future Child. **19**(1), 157–183 (2009)
12. Verner, I., Ushin, I., Korchnoy, E.: Learning physical fields through operating robot movements: a case study. In: Jamshidi et al. (eds.) Robotics, Manufacturing, Automation and Control, vol. 14. TSI Press, Albuquerque, NM, pp. 383–388 (2002)
13. Korchnoy, E., Verner, I.: Characteristics of learning computer-controlled mechanisms by teachers and students in a common laboratory environment. Int. J. Technol. Des. Educ. **20**(2), 217–237 (2008)
14. Cuperman, D., Verner, I.: Learning through creating robotic models of biological systems. Int. J. Technol. Des. Educ. **23**(4), 849–866 (2013)
15. Verner, I., Hershko, E.: School graduation project in robot design: a case study of team learning experiences and outcomes. J. Technol. Educ. **14**(2), 40–55 (2003)

16. Verner, I.M., Revzin, L.B.: Characteristics and educational advantages of laboratory automation in high school chemistry. Special focus paper. Int. J. Online Eng. **7**(S1), 44–49 (2011)
17. Verner, I.M., Revzin, L.B.: Automation of manual operation in a high school chemistry laboratory: characteristics and students' perceptions. The Chem. Educ. **15**, 141–145 (2010)
18. Goldin, G.A., Epstein, Y.M., Schorr, R.Y., Warner, L.B.: Beliefs and engagement structures: Behind the affective dimension of mathematical learning. ZDM Math. Educ. **43**, 547–560 (2011)
19. Verner, I.: Characteristics of student engagement in robotics. In: Omar, K., et al. (eds.) FIRA 2013, CCIS 376, pp. 181–194. Springer, Heidelberg (2013)
20. Tortosa, M.: The use of microcomputer based laboratories in chemistry secondary education: present state of the art and ideas for research-based practice. Chem. Educ. Res. Pract. **13**, 161–171 (2012)

Breeding Robots to Learn How to Rule Complex Systems

Franco Rubinacci, Michela Ponticorvo, Onofrio Gigliotta and Orazio Miglino

Abstract Educational robotics has been extensively used to teach hard skills such as computer science, computational thinking and coding because traditional robotics is the outcome of analysis, design and programming. Other approaches to robotics, namely evolutionary robotics, open the way to reflection on emergence, self-organization, dynamical systems. As these issues are relevant in present days society, we propose a robotic laboratory where children are trained to rule complex systems. In particular, the integrated hardware/software system BrainFarm, that allows to evolve and train virtual robots and then test them in physical environments, is employed to train these skills and a successful experience in informal context is described.

Keywords Dynamical and complex systems · Robotics lab · Evolutionary robotics

1 Introduction

Cultivating, breeding, training and teaching are activities that men employ to rule and shape nature. Farmers, breeders, teachers and parents have always used their ability to observe, follow and address the development and evolutionary pathways of living beings, be it, plants, animals or children. Even if it is clear that a botanist and a teacher have different expertise and adopt different techniques, they share the ability to establish a connection and a dialogue between two autonomous entities. They follow organisms' history and intervene to correct, inhibit, facilitate behaviours or morphologies. In this process the man does not control nature completely: every organism has a certain amount of autonomy and therefore it can not be controlled, but rather looked after.

On the opposite side, technology has developed effective techniques to build machine which are completely different from the ones we have just cited and which

F. Rubinacci · M. Ponticorvo (✉) · O. Gigliotta · O. Miglino
Department of Humanistic Studies, University of Naples "Federico II", Naples, Italy
e-mail: michela.ponticorvo@unina.it

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_13

are adopted by breeders. Machines are the final outcome of analysis, designing and programming: no space for autonomy is left to machines, no unforeseen is allowed.

For many years breeders and engineers have had nothing in common. This is no longer true as there are nowadays organisms that are designed and machines that are bred. Biotechnologies allow to design living organisms to a certain extent, whereas new approaches to technology have opened the way to emergence, self-organization, dynamical systems. It can be extremely difficult to design a priori an effective solution to some problems which are unpredictable and require adaptation, change, flexibility. In these cases it is necessary to provide machines with adaptation mechanisms.

In robotics many approaches have been proposed in this vein, such as adaptive robotics [1], biorobotics [2], and evolutionary robotics [3, 4]. On the educational side, robots have been extensively and successfully used to teach hard skills, related to STEM [5]: in this case robots are designed, programmed and implemented. Robots are machines which undergo an engineering process.

But, if the robot is seen as an organism that must adapt to the environment it operates in relying on its motor apparatus and the stimuli coming from the external and internal environment, it is possible to consider it as a complex system to be ruled. The present society requires children, the future citizens, to deal with self-organizing entities, with dynamical systems, with emergence. In other words, together with hard skills related to precise expertise, everyone will have to master transversal competencies, soft skills, including complexity management.

To help achieving this goal, breeding robot can be the main activity of a robotic laboratory to train to manage complex systems. In what follows the robotic system BrainFarm is described together with a successful experience in informal, noncurricular educational context.

2 A Robotic System to Breed Robots: BrainFarm

BrainFarm and its prequels BreedBot and BestBot [6, 7] are integrated software/ hardware platforms that allow players, even without any programming or computer skill, to breed, within customizable virtual worlds, artificial organisms that can be downloaded onto real robots (Fig. 1).

The breeding is implemented through a user-guided genetic algorithm [8], based on Interaction Evolutionary Design [9]. Evolutionary robotics aims at developing robots taking inspiration from evolution in biology. Small populations of robots undergo an evolutionary process to adapt to a particular environment and accomplish a task. The robots are reinforced with reinforcement learning algorithms [10] which model learning theories developed in psychological research. Interaction Evolutionary Design allows the user to realize objects continuously interacting with a software that proposes many variants of the object itself.

Therefore the software side of BrainFarm shows users a population of nine wheeled robots. Every robot is provided with some infrared sensors, to detect nearby obstacles, and two motors that control wheels speed. A differential drive system

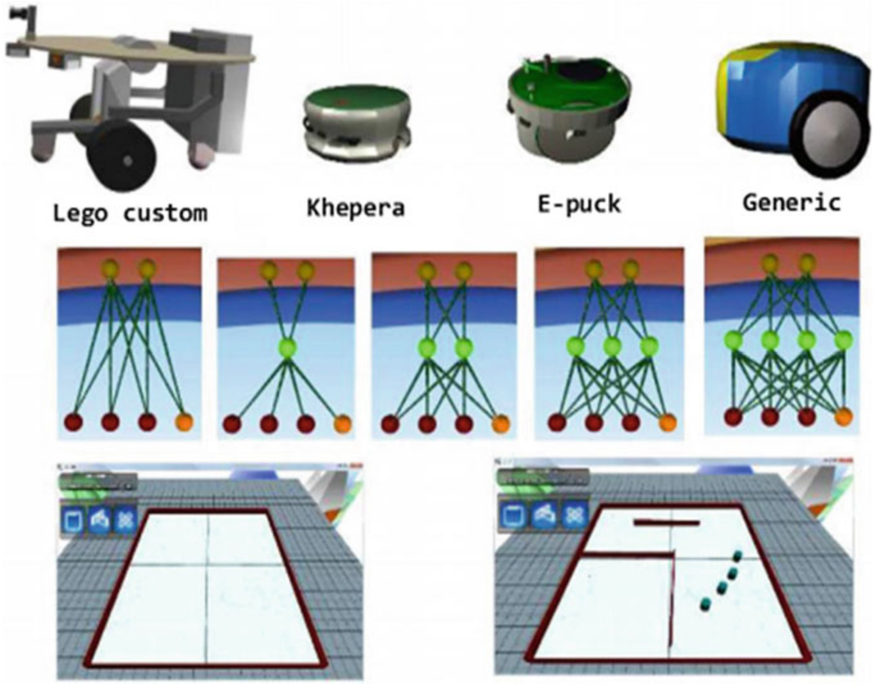


Fig. 1 The BrainFarm interface with custom neural networks

allows robots to steer in any directions. Each robot is controlled by a simple feedforward neural network, representing an artificial nervous system [11]. The input layer consists of infrared sensors and two motor context units that are simply two relay units of the previous motor activation (all inputs are normalized between 0, 1). The input neurons are connected to the output layer made up of two motor neurons that control the right and left motor of the robot. Robots speed is updated according to motor neurons activation. The neural network parameters are in turn encoded in a genetic string that will undergo an evolutionary process [12] guided by either the users (artificial selection) or the machine (automatic selection). In this latter case the player can anyway manipulate the relevant evolutionary variables.

The player can affect directly the evolutionary pathway acting as a breeder that selects the preferred agents or observe the possible evolutionary outcomes depending on different conditions. In this latter case, by manipulating directly the parameters related to the genetic algorithm the player can understand the underpinning dynamics and experience different evolutionary pathways in a controlled environment. Moreover BrainFarm allows to design the robots brain architecture. Users can use a simple feedforward network or more complex architectures to control robots behavior, as shown in Fig. 1. This allows to introduce users to brain structure and dynamics, through artificial neural networks models.

These platforms have been used to teach evolutionary biology [13], that is to say a hard skills, but they are fit to soft skills training too. In particular BrainFarm can be a gym where children can receive a training to complex systems management. An experience about this approach is introduced in next section.

3 The BrainFarm Lab with Children

In this section we describe a successful laboratory experience that was held in Naples, at *Citta' della Scienza*. This is a cultural initiative to promote and popularize scientific knowledge. *Citta' della Scienza* has a multifunctional structure with an interactive scientific museum and a training centre. In this latter centre many initiatives allow children to take part to laboratories in an informal context.

3.1 Partecipants

The BrainFarm lab lasted from January 2016 to February 2016 and involved 10 classes of about 25 student accompanied by 2 teachers.The BrainFarm lab was attended by 245 children with an average age of 11 years. 20 teachers took part to the lab together with students.

3.2 Lab Scheduling

The BrainFarm lab was arranged as depicted in Fig. 2: first of all a pre-lab questionnaire was administered both to teachers and students to assess their expectations about the laboratory. In particular teachers and students expectations about the chance to learn complex systems management were investigated. The lab itself lasts 75 min: after a theoretical introduction about robotics, algorithms and interaction design, children can build a simple rover bot according to a suggested morphology. Then the task to be accomplished is introduced. It is a navigation task that foresees a starting point, some areas to be reached and the target reaching. The environment is built in the physical world and then reproduced in the software.

The group is divided into 4 sub-groups that must interact with the BrainFarm software to select and evolve the best robots according to the Interactive Evolutionary Design introduced above. In other words they select the best robots on the screen and allow them to reproduce. This way a new generation of robots, similar to the selected ones but with some differences, can be observed and undergo the selection process. The robot morphology does not change along the process, but the parameters of the artificial neural networks are modified.

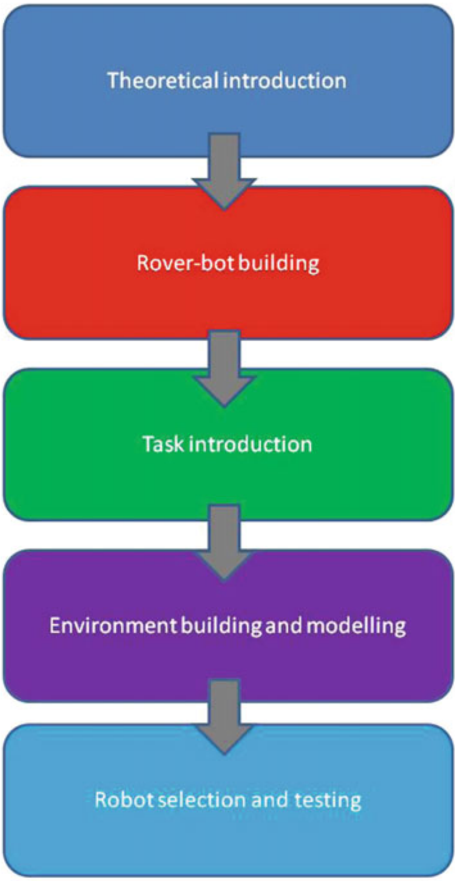


Fig. 2 The BrainFarm lab schedule

The students then test their selected robots by transferring the simulated control system into the physical robot in the physical environment. They act as breeders that allow to reproduce only the best exemplars, according to the proposed solutions.

At the end a post-lab questionnaire is administered to teachers and students to verify if the lab was effective, in their opinion, to train about complex system managing.

3.3 Comments on BrainFarm Lab

In this section we will report some preliminary results about the lab experience at *Citta' della Scienza*. Teachers and students liked the lab very much, 84 % of students rated the lab as very interesting on a Likert scale from 'not interesting at all' to 'very interesting'. The great majority of students appreciated the chance to work with robots and breed them (82 %). Students do confirm tat the observing and ruling the evolutionary process increased their interest in science, technology, designing and in biology, especially related to evolution, development and adaptation. In fact on a Likert scale from 0 ‘I do not agree at all’ to 5 ‘I completely agree’, students rated, on average, their increase of interest in cited discipline as follows: science 4.2; technology 4.5; designing 4; biology 4.8. They also confirmed that an appealing aspect was related to the unforeseeable behaviours displayed by robots in interaction with the environment 75 %. They had to carefully observe single robots behaviours to choose the best exemplars and this stimulated their ability to address solutions without designing them. Teachers answered questions about soft skills, and they confirmed that communication between students was stimulated (90 %), problem solving and group working were favoured (83 %), during the lab experience, in respect to routine activities.

The BrainFarm lab allowed students to increase the ability to communicate well, to develop good relations with others and to find shared solutions. Moreover students learned that it was possible to make mistakes and that learning from mistakes is good for everyone. The user acts a breeder, thus learning in a safe context to build an expertise in selecting solution and using this knowledge to rule a complex system.

References

1. Ziemke, T.: The construction of reality in the robot: constructivist perspectives on situated artificial intelligence and adaptive robotics. Found. Sci. **6**(1–3), 163–233 (2001)
2. Webb, B., Consilvio, T.: Biorobotics. Mit Press (2001)
3. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-organizing Machines. MIT Press (2000)
4. Ponticorvo, M., Walker, R., Miglino, O.: Evolutionary robotics as a tool to investigate spatial cognition in artificial and natural systems. Artif. Cogn. Syst. 210–237 (2007)
5. Eguchi, A.: RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition. Robot. Auton. Syst. **75**, 692–699 (2016)
6. Gigliotta, O., Miglino, O., Schembri, M., Di Ferdinando, A.: Building Up Serious Games with an Artificial Life Approach: Two Case Studies. In Evolution, Complexity and Artificial Life, pp. 149–158. Springer, Berlin (2014)
7. Miglino, O., Gigliotta, O., Ponticorvo, M., Nolfi, S.: Breedbot: an evolutionary robotics application in digital content. Electron. Libr. **26**(3), 363–373 (2008)
8. Oladiti, B.T.: User-Guided Evolutionary Algorithms (1997)
9. French, M.: The Interplay of Evolution and Insight in Design. Evolutionary Design by Computers, 77 (1999)
10. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. 237–285 (1996)
11. Patterson, D.W.: Artificial Neural Networks: Theory and Applications. Prentice Hall PTR (1998)
12. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. **3**(2), 95–99 (1988)
13. Miglino, O., Rubinacci, F., Pagliarini, L., Lund, H.H.: Using artificial life to teach evolutionary biology. Cogn. Process. **5**, 123–129 (2004)

A Thousand Robots for Each Student: Using Cloud Robot Simulations to Teach Robotics

Ricardo Tellez

Abstract One of the main problems when teaching robotics is the lack of robots for all the students in the class due to its costs and difficulty to maintain. In this paper, we analyze the advantages and drawbacks of using simulations for teaching robotics instead of real robots, as a solution to that problem. We describe a simulation tool based in the cloud that allows the simulation of complex robots off the shelf by using only a web browser as the base system for teaching robotics. Finally we provide the description of a teaching protocol for minimizing costs and troubles while maximizing students experience, which makes a combined use of simulations and real robots.

Keywords Teaching robotics · Robot simulations · Web application

1 Introduction

Robotics is a fashion subject, that is in an exponential growth all over the world. New companies about robotics are created every week and the forecast indicates that it is going to increase even more [1]. Hence, there is an increasing need of more prepared engineers in this subject.

Teaching robotics requires to teach in the most optimal way in order to transfer a knowledge that advances quickly, and to engage students into it. Fortunately, at this moment in time, we have access to a large base of commercial robots, schematics of open source robots [2], and free software [3]. All of that makes learning robotics easier than ever. However, the own existence of such large number of robotics options, makes very difficult to teach it, because making the learning experience work for all the students of the class can be, first, very expensive, and second, very complex.

R. Tellez (✉)
The Construct Sim, San Francisco, USA
e-mail: rtellez@theconstructsim.com

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_14

We identify the main problems of teaching robotics as the following:

1. Robots and components for robots are expensive. Hence it can be very expensive for the school to provide those robots or parts to each of the students.
2. Even if the school has access to the robots or parts, it takes a long time to correctly make work a given robotic setup. Robotics is a difficult subject, and requires the proper functioning of a large amount of different parts. If the student has to build all those parts and make them work it is going to take time. Furthermore, the number of problems exponentially increase with the number of students in the class.
3. Finally, setting up a robotics development environment requires a huge effort for the students. Teachers can design the best experiments with robotics ever, but making those run in the students own environments (or even in the environments the school provides) is very tricky. Students have access to different types of computers and operating systems. In the same line, schools have different computer configurations configurations, old computers, etc. Hence, even if the teacher achieves to make the exercises work in her system, making the exercises work for all the students is a different and more complex task.

In order to solve the first and second problems we propose the use of robot simulators for teaching robotics. In order to solve the third problem we propose the use of web based simulations in the cloud.

2 Using Simulations for Teaching Robotics

The use of robot simulations for teaching means that, instead of using a real robot to make students learn about robotics, the teacher uses a computer program (the simulator) that shows the robots on the computer screen as if it were the real robot. The simulated robot is called a model of the robot. There are many different simulators available, each one with its own characteristics and working flow. Depending on the simulator used and of the model quality, the model can look and behave more or less close to the behavior of the real robot (see Fig. 1 with real Nao robot and Nao simulation model for Webots simulator).

In order to start simulating a robot, the simulator program must be installed in the computer of the user previously to start working with it. It usually requires specific setups or configurations (for example, a given operating system in the computer, some specific libraries installed, etc.) prior to the installation. Those requirements make the installation process a little complex in most of the cases.

Once a simulator has been installed in a computer, the teacher must create the model of the robot that she wants to use in her classes. Another option is to get the model from an already created source. There are different repositories on internet providing models already made for very famous robots, even if in most of the cases, make those models work on the students computers represents a challenge in itself.

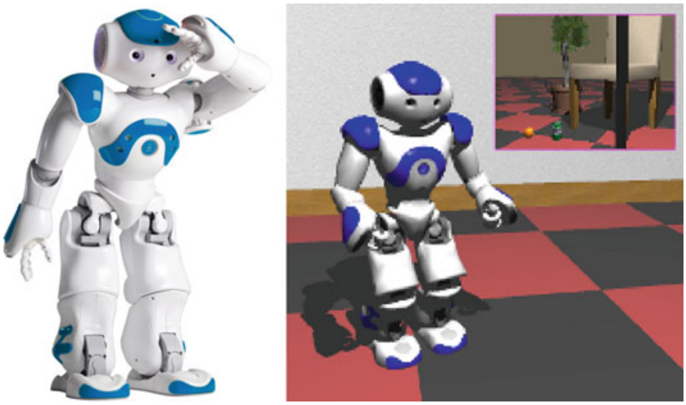


Fig. 1 The real Nao robot and its simulation model of Webots simulator

Users always have the option of building their own models for existing robots, or for other robots that they invent.

Once the robot model is available, it is required to create a simulated environment for it. For example, the Nao robot of Fig. 1, is in a main room environment. The whole thing including the robot model and the environment is called a simulation. The simulation, is the program is executed inside the simulator and describes what must be shown on the screen by the simulator (that is, the robot in its environment).

The robot model must have a programming interface that allows users to command the simulated robot, reading its sensors and sending commands to its actuators. For example, in the case of Nao robot, the programming interface allows users to capture data from the (simulated) camera that the robot has in its head, or send commands to the motors of the joints and make the robot walk.

Once a model and its environment are created inside the simulator, teacher and students can program it as if it were the real robot using the programming interface. When the simulator executes the created program, the resulting behavior of the robot is shown on the screen.

Latest simulators tend to present to the user the same kind of programming interface that the real robot has. This fact represents a huge improvement from old robotics systems, since a program created for the simulated robot can be used later in the real robot without modification, and make the real robot behave (more or less) the same as the simulated robot.

Finally, to say that there exist a lot of different simulators available, each one with its own specifics. Some simulators are mainly focused on industrial robots, others on service robots. Some are based on standards and some have their own protocol. Deciding which one to use may be key for the teaching at hands. Table 1 contains a table of the most used simulators for robotics.

Table 1 List of most popular simulators, their intended scope and their availability for different operating systems

Simulators	Operating system	Scope
Gazebo	Linux, MacOS	Service and industrial robots
Webots	Windows, Linux, MacOS	Service and industrial robots
V-rep	Windows, Linux, MacOS	Service and industrial robots
UsarSim	Windows	Service and industrial robots
Morse	Linux	Service robots
RobotStudio	Windows	Industrial robots
Virtual robotics toolkit	Windows, MacOS	LEGO and VEX robots
Visual components	Windows	Industrial robots
Robot virtual worlds	Windows	LEGO, VEX and TETRIX robots

2.1 Advantages of Using Simulators

Using a simulation in front of a real robot has several advantages:

1. High reduction of the cost. Real robots tend to costs between hundreds and thousands of euros. Simulations instead can start from zero to a hundred, depending on the requirements and quality of the simulator.
2. Each student has access to her own (simulated) robot. Having a real robot for each student is impossible even in the best schools, because of the prices. The result is that the students must group in order to access the real robot and test their algorithms. This results in a very poor learning experience for the student. Instead, when using a simulator, each student can have to the simulated robot by using only a computer.
3. Students cannot break the robot by doing errors. If a real robot is used, a simple mistake in the code can make the robot crash or fall, breaking its parts. With simulations, student's mistakes cause no harm.
4. Access to the best robots of the world (see Fig. 2). The best robots of the world, the ones being used now in the best research labs of the world, have a simulation model available. By using a simulation, students can access those robots and learn from the results of the best labs of the world.
5. Access to many different environments for the robot. Even if the school manages to have a real robot for each student, it is very unlikely that will be able to provide interesting environments for those robots (basically because you have to build those environments). If you use a simulator, you can place the robot in any environment you can think of. For instance it is possible to have a robot on space, deep water or nuclear reactor, and test the students programs for the robots in those environments.

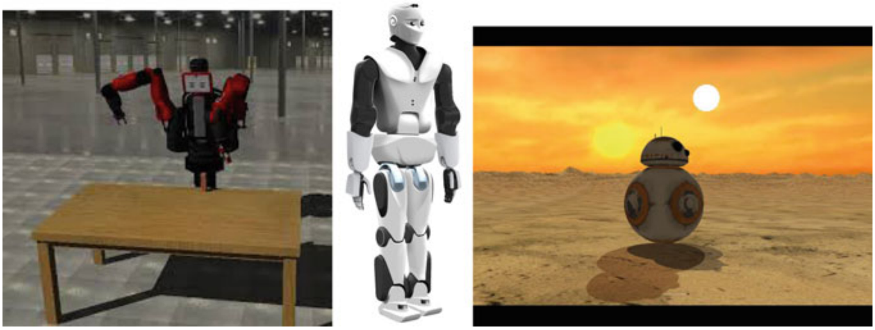


Fig. 2 A simulation of Baxter (from Rethink robotics), Reem-C (from Pal robotics), and BB-8 (from Disney) robots, some of the most advanced robots in the world

2.2 Drawbacks of Using Simulators

Using a simulator for teaching, has a series of drawbacks in front of real robots:

1. No learning about real hardware. By concentrating the student on the simulation, she learns all the inner workings about simulations, a thing that is very different from the inner works of real robots. Students don't get this special learning that is obtained by working with real stuff, and all the problems that are associated to real robots.
2. Learning with simulators is mostly about software, with almost no learning about mechanics nor electronics. 90 % of simulators are concentrated on learning how to program a robot to do something. There is no learning about how to build the mechanical parts of the robots or the electronic systems inside the robot that run the control software.
3. Difference between the behavior of real and simulated robot. Even for the best simulators, the models of the robots are not perfect. Hence, the behavior of the simulated robot varies from the behavior of the real robot. How this variation is important or not depends on the level of programming one wants the students to concentrate at. If the level of programming is at that of the controller, then there is usually a large difference between real and simulation. If instead the focus at the level of functionality, the behaviors are very similar to real robots.
4. Simulation are less engaging for students. Making a simulated robot grasp an orange from a table is not as exciting as making the real robot do it. Hence, simulations do not have the same appealing as real robots. However, this engagement highly depends on the implication of the teacher and her ability to create really engaging simulation examples or exercises.
5. Many of current robot simulators are tricky to install, set up and have them working in different equipments. When using a simulator, there is a previous work that needs to be done by the teacher, which requires installing the simulator in the computers that are going to be used, if they belong to the school. If students'

computers are going to be used, all kinds of errors and problems appear just trying to install, for which the teacher must provide some kind of support.

6. Creating the robot model at present is not a simple task and requires a lot of previous work to create it. This situation is at present quickly improving, because there is an emerging market for web pages that provide already made complex robot simulations to be used off the shelf [4, 5].
7. If several simulators are needed (because of different levels of proficiency of the students, or because teaching different functionalities are required) then the setting up work multiplies and becomes more complex. Of course, teachers must provide the simulation for each of the setups which is not straight.

3 Using Web Based Simulations for Teaching

Given some of the drawbacks presented in the previous section, we suggest the use of web based simulators for teaching instead of desktop based simulators. The difference between desktop and web simulators is that the former requires to be installed in a specific computer (with all the drawbacks indicated above), and the later requires no installation at all. While desktop simulators require the user to install the simulator in the user desktop or laptop computer, the web simulator is just a web page that the user can access with her account using only a web browser. Users can use any browser, which means that they can simulate from any computer at any location, including from home, school of internet cafe.

3.1 Advantages of Web Based Simulators

These are the advantages of using a web based simulator:

1. Do not need installation nor maintenance. Installation and maintenance of the simulator is performed by the web portal that provides the simulator. Hence, neither the school nor the students have to deal with this unrelated task. This is specially convenient for some schools where the policy of installation of software is very strict and requires a long bureaucracy process.
2. Teachers can be sure that their exercises are going to work as they have designed them. Since all students and teachers are using the exact same simulation system, the results that are going to be obtained are exactly the same. Teacher can prepare exercises and show steps to follow that, if correctly performed, will produce the expected results in the simulated robot.
3. Students cannot break the simulation program by making errors or miss use of the simulation. If students make mistakes and something goes wrong crashing the simulation, they just can relaunch the web simulation again and start again from the initial conditions. Performing that error in desktop computers is one of the errors that consumes most of the time when teaching robotics, since students

perform errors that destroy the simulation system in their computers. This may require even to reinstall the whole operating system. With web simulators students cannot break any system, what makes students more prone to experiment and try new programming ideas.

4. Different simulators available which allows entrance at different level of complexity. A web simulation system can provide different simulators at the same portal, each one concentrated in teaching a specific subject (for instance, simulators concentrated in robotics arms, or simulators for humanoids or simulators for evolutionary robotics), or different levels of proficiency (for high school students, for university students, etc.).
5. Students can use any type of computer. In a class, there are all kinds of students, each one using their preferred operating system. As can be seen in Table 1, not all the simulators can work in all the operating systems. With a web simulator, this problem fades away since the only required thing is a web browser, which is available for all types of operating systems.
6. Students and teachers can work from anywhere. Since the only requirement is to have access to internet, teachers and students can actually be anywhere while doing their simulations. This opens a huge door for remote teachings.
7. Students can cooperate with their mates while working on the simulation. Web simulation is by nature a collaborative thing, at the contrary of desktop simulations. Hence, differently from desktop simulators, web simulators allow the work of different people at the same time on the same simulation. This is very useful when students need to cooperate to program the different parts of a robot, or when the teacher needs to see where the student is stuck and help her get the solution.

3.2 Drawbacks

As any system, web simulations have also drawbacks:

1. Users need a fast enough internet connection. Web simulations are run through an internet connection, and since the simulations tend to be complex, it is required to have an internet connection with fast speed.
2. By being on the cloud, web simulation inherits all the security problems that any cloud system has. It is very difficult to ensure that the data stored in the cloud will not be accessed by unauthorized people.

4 A Case Study: The Construct Web Simulation System

As a solution for web simulation we propose The Construct. The Construct is a web platform in the cloud that provides a large list of simulators ready to be used by means of a web browser. It is a web based simulator with all the advantages (and drawbacks) explained in the previous section.

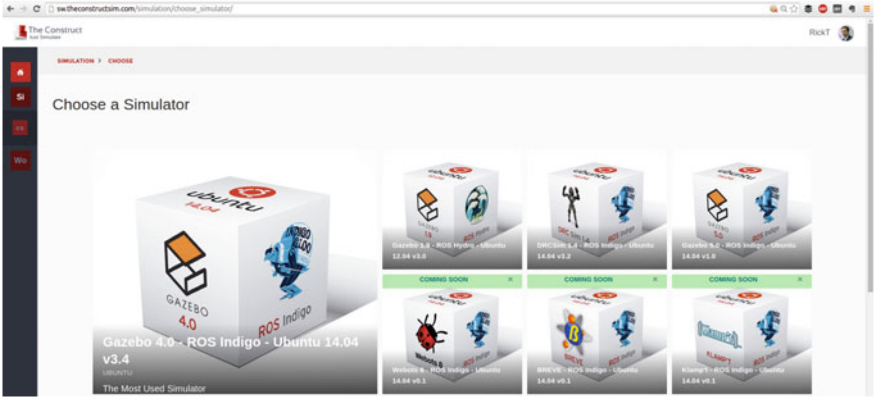


Fig. 3 A view of the simulators available at The Construct

The Construct allows to execute simulations off the shelf by using WebGL visualization. Users do not have to install anything, not even in their browsers, since all modern browsers already support WebGL based code. When the user access the platform using her preferred web browser, she has access to a list of different preconfigured robot simulators (Fig. 3).

Once the user access the portal, she can select which simulator to start simulating with. Selection of the simulator depends on the type of work she is going to do, and the level of complexity of the simulations. For instance, for beginners, it is easier to start using Webots simulator [6] because it is more simple in terms of interface and tools. However, if you want to learn about the latest robotics technologies like ROS [3] you may need to use the Gazebo simulator [7]. Finally, if you want to develop robotics with deep learning you should use the DRC+CUDA simulator [8].

The simulators available in The Construct are in fact existing desktop simulators. This implies that simulations created with the desktop versions of the simulator are compatible with the web version, and vice versa. Hence, teachers and students can upload to The Construct their already existing simulations and run them in the web. The opposite is also possible.

The Construct only needs a WebGL enabled browser. Most modern browsers are. Officially supported are Safari, Chrome, Firefox. This means that it can be used with any device that has one of those browsers, including Linux, Windows, MacOS, or even tablets and smartphones.

Apart from showing the simulator on the web browser, The Construct incorporates a series of very useful features for teaching robotics, that make the system go far beyond its desktop versions.

4.1 Sending Simulation Files to Students

When a teacher has created a simulation for their students, she can directly share the files that compose the simulation with all her students with a single click, without having to send files through email, or storing the files in some place of the internet. With a single click, the file selected is sent to a list of students provided by the teacher. From that moment, the students of the list can run the exact simulation that the teacher created, each student in her own environment free to experiment without interfering with the other students simulations. The opposite is also true: the students can send their simulations to the teacher from within The Construct and be sure that their simulations will work for the teacher as they expected.

4.2 Sharing Simulation with Teacher

If a student has a problem and does not know how to continue, she can request help from the teacher, by sharing her running simulation with the teacher. When a running simulation is shared, both the student and the teacher are seeing the same running simulation. This implies that the teacher can see what is failing in the simulation and programs of the student, and help her understand the error. The teacher has complete access to the simulation of the student and can modify it or suggest new ways of continuing.

This same mechanism of sharing can be used between students to collaborate between them in the programming of a simulated robot. In the same sense a Google Docs allows different people to write over the same document, sharing a simulation allows different people to work over the same simulation. It is just a matter of selecting the option of sharing and specifying the list of users that the owner of the simulation wants to share with.

After the collaboration is finished, the owner of the simulation has the right at any time to close the sharing session and forbid the access to her simulation.

4.3 Integrated Development Environment

Neither students nor teachers need extra programs than the web browser to simulate the robots and create the control programs for the simulated robot. A full environment is provided including IDE, web console and also a Python web environment. Hence, the full learning can be done using only a web browser.

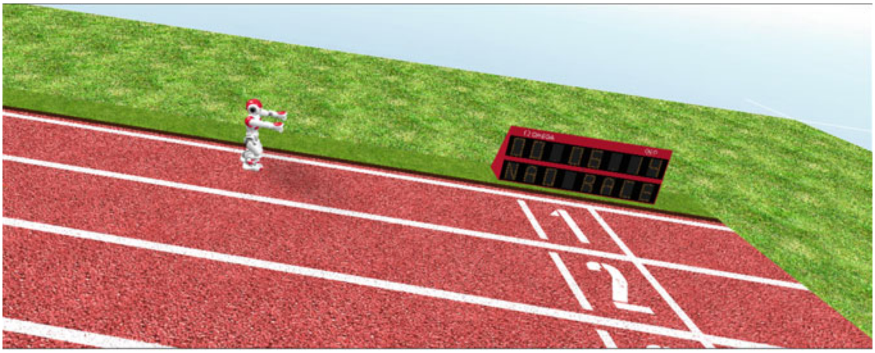


Fig. 4 Simulation contest of Nao robot running the 10 m

4.4 Engage Students Through Simulation Contests

One way to encourage students and robotics research is by doing contests where the participants have to compete against each other. For example one of the most important robotics simulation competition was the Virtual Robotics Challenge [9] where the simulation was web based and involved the best robotics teams of the world to control a human size humanoid robot on a nuclear disaster area.

This kind of competition can be easily done using web simulations at The Construct, like for example the Robto Race to Hawaii contest based on Nao robot race, where students around the world should make a Nao robot walk for 10 meters as fast as possible (see Fig. 4).

5 Proposing an Optimized Robotics Teaching Environment

One of the main criticisms to using simulations to teach robotics instead of real robots is that students get a distorted vision of the field and separate themselves from the real robots, by concentrating too much on the simulations instead of real robots. One way to solve this problem without incurring into too much costs is to combine simulations with real robots.

Modern simulators work based on a hardware abstraction layer (HAL) [10]. A HAL is a framework for a robot in which the commands sent to the motors or sensors of the robot do not differentiate between the real robot or the simulated robot. Usually, the HAL is provided by the company that builds the robot. Sometimes, the robotics community creates this HAL for interesting robots that do not come with it.

Other times, when a robot does not provide such HAL, the simulator may provide instead a cross compilation facility [11]. This is the case for example of some of the robots provided by Webots simulator. In the case, for instance of Aibo robot, it does not have such abstraction layer due to its low CPU computer and being an old robot.

Webots provides a mechanism that allows the translation of the code you use in the simulation of Aibo robot to code to be executed in the real robot. The result is that, the same things that the robot does in the simulation with the user program will be done by the real robot when using the cross-compilated code [12].

Having this framework available, either through HAL or cross-compilation, a new teaching environment that optimises costs and maximizes students experience can be devised. We call this environment the Optimized Robotics Teaching Environment (ORTE). It works as follows:

1. The teacher selects the real robot to use and buys it. Only one or two units of the real robot may be necessary (depending on the amount of students per class and the resources available).
2. The teacher creates a simulation of the robot which mimics the way the real robot will be used for the class. It is very likely that a simulation for that robot and environment already exists in the net, hence it is suggested to perform a search for it. Getting an already made model of the robot can save many days of work.
3. The teacher distributes the simulation among students and specifies what it is required to do (for instance, make the robot recognize an orange, make it walk up to the table, etc.).
4. Students create their control programs for the simulated robot in order to accomplish the assigned task.
5. Once a student has achieved a program in the simulation that performs the task, she can transfer her program to the real robot making use of the HAL or the cross-compilation, and check if it works the same way in the real robot. Chances are not. Hence the student returns back to simulation to analyze why not and devise a new strategy.
6. Keep repeating fourth and fifth points until fifth is accomplished.

One example of successful use of the ORTE is the MIT course MIT 2.166 of 2016 on autonomous vehicles (class named Duckietown [13]), where students have to build their real robots using arduino boards and use the simulation of those robots in The Construct to build and test control programs (Figs. 5 and 6 show the real and simulated Duckiebot model and real and simulated environment).

6 Conclusion

Simulations are a powerful tool for teaching robotics, since allow the creation of cheap robotics environments, and access to the latest (simulated) robotics technologies at a very low cost.

Simulations can be simplified for teachers and students by making use of web based simulators. Web based simulators universalize the access to such software because allow anybody use them with any device with a fraction of the cost of the desktop simulator.

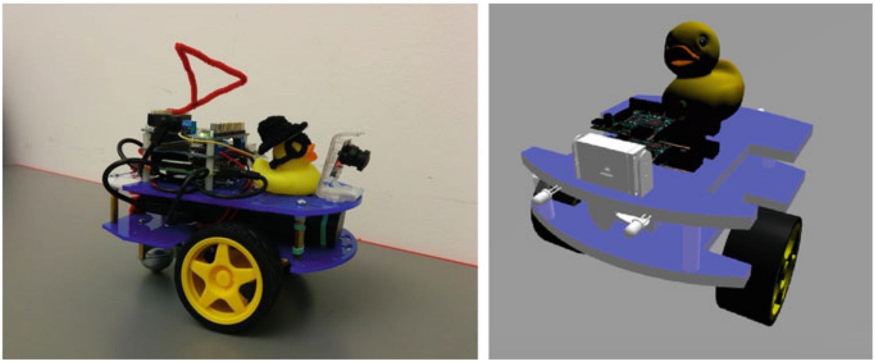


Fig. 5 Real Duckiebot and its simulation model

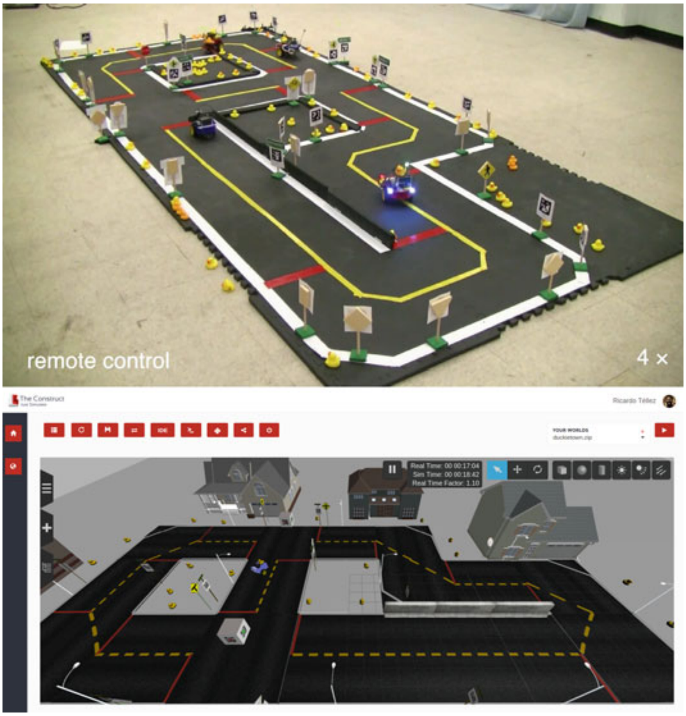


Fig. 6 Real Duckietown environment and simulated environment using The Construct

Finally, simulations can be integrated in a complete learning system that includes real robots in the path, minimizing costs and maximizing usage and students experience (ORTE method).

Acknowledgments We would like to thank all the Duckietown team for making it possible and specially Andrea Censi for making us part of the project.

References

1. Silicon Valley Robotics: Service robotics case studies in silicon valley. Tech Report (2015)
2. Poppy humanoid robot. http://www.poppy-project.org
3. Robotics operating system. http://www.ros.org
4. Simulating Husky robot: the easy way. http://www.theconstructsim.com/simulating-husky-robot-the-easy-way
5. Simulating Tiago robot: the easy way. http://www.theconstructsim.com/simulating-tiago-robot-the-easy-way
6. Michel, O.: Webots: professional mobile robot simulation. J. Adv. Robot. Syst. (2004)
7. Koenig, N., Hsu, J.: The many faces of simulation: Use cases for a general purpose simulator. IEEE International Conference on Robotics and Automation (2013)
8. Hsu, J.M., Peters, S.C.: Extending open dynamics engine for the DARPA virtual robotics challenge. In: 4th International Conference, SIMPAR (2014)
9. Agero, C.E., Koenig, N., Chen, I., Boyer, H., Peters, S., Hsu, J., Gerkey, B., Paepcke, S., Rivero, J.L., Manzo, J., Krotkov, E., Pratt, G.: Inside the virtual robotics challenge: simulating real-time robotic disaster response. IEEE Trans. Autom. Sci. Eng. **12**(2) (2015)
10. Jorg, S., Tully, J., Albu-Scheffer, A.: The hardware abstraction layer—supporting control design by tackling the complexity of humanoid robot hardware. IEEE International Conference on Robotics and Automation (2014)
11. Michel, O., Rohrer, F.: The Rats Life Benchmark: Competing Cognitive Robots RSS (2008)
12. Holh, L., Tllez, R., Michel, O., Ijspeert, A.: Aibo and Webots: simulation, wireless remote control and controller transfer. Robot. Auton. Syst. **54**(6), 472–485 (2006)
13. Duckietown MIT 2.16. http://duckietown.mit.edu/

Part IV
Technologies for Educational Robotics

Networking Extension Module for Yrobot—A Modular Educational Robotic Platform

Michal Hodoň, Juraj Miček and Michal Kochláň

Abstract The contribution describes a modular educational robotic platform Yrobot and its networking extension module. This module is designed for wireless communication among multiple Yrobot platforms as well as for interfacing with smart devices and computers. The networking module enables to verify various wireless communication standards and technologies in ISM band. In the text, the motivation for developing of such extension module is described as well as circuit solution implementation and firmware attributes for the module.

Keywords Networking extensions · Yrobot · Educational robotic platform

1 Introduction

Mobile robotics systems for educational purposes come still more and more to the foreground especially in terms of modern teaching. This progress goes hand-in-hand with OpenHW systems advancements, which enable a wide community of people to enter the world of computer engineering. In [1], the authors proposed lab work for learning fault detection and diagnosis, training the skills important for engineering education in mechatronics. An approach in [2] is based on the pushing of students to design and test their own original circuits and software code by modifying, extending or expanding the sample circuits and example code described in the lecture notes, in order to keep students highly curious, motivated and engaged in self-regulated learning. The work in [3] presents the design of an open educational low-cost modular and extendable mobile robot based on Android

---

M. Hodoň (✉) · J. Miček · M. Kochláň
University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia
e-mail: michal.hodon@fri.uniza.sk

J. Miček
e-mail: juraj.micek@fri.uniza.sk

M. Kochláň
e-mail: michal.kochlan@fri.uniza.sk

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_15

and Arduino to be used as an educational tool in labs and classrooms. Other low-cost but open-source robotic platform AERobot usable for introductory programming and robotics teaching was described in [4]. Further interesting robotic platforms as well as they use in education were described for example in [5–9]. Beside the tangible platforms, robotic simulation environments took place in an educational process as well, e.g. [10, 11].

Yrobot high-school education kit was for the first time introduced at 2014 RAAD conference. From that time, already after 18 months, the robot was successfully integrated into the educational process in more than 20 high schools in Slovakia. During 2015, we decided to extend the kit with additional application modules to target an actual problematic of wireless communication and derived areas, such as Internet of Things; Collaborative Signal Processing; Distributed Signal Processing; etc. For that reason, Yrobot modules “Y-WiFi”, “Y-BlueTooth” and “Y-ZigBee” were introduced. These modules can be easily installed within the platform, where provided functionalities, which can be easily translated into educational process. Especially in the popular fields such network administrator or network specialist.

To understand the concept of Yrobot, we have to mention an effort of Volkswagen Foundation, which was supporting the project continuously from 2013. The idea behind was to develop an educational kit that can be used for the purposes of IT education among the Slovak high schools. Original intent of authors was, by using of a simple technical device, to increase the motivation of secondary-school-students in the study of technical fields. Among the others, especially to arise their interests in information technology. The concepts of the system, its features and functions, as well as initial results from deployment in teaching were presented in [12, 13].

As supporting activities, different workshops, where particular teachers got in first touch with the platform, were realized. Furthermore, two regional competitions “YrobotCup - lets program the robot” were organized for the students of Žilina and Bratislava region. In these contents, the students could compare their knowledge with other competitors what could motivate them in other work with Yrobot. To facilitate the work with Yrobot, the textbook where basic principles together with programming examples were described by the usage of simple, friendly and easy-to-understand way, was published [14].

As a reminder to be mentioned [12, 13]—the robotic platform has the nature of open source hardware with presumes on the development of interesting applications (hardware and software modules) directly by the secondary school students. In the figure below (Fig. 1), the platform with textbook is shown.

To invent an interesting and non-traditional application with the design of appropriate HW and SW means is a very complex task. The successful solution implies a good knowledge of several disciplines (sensory, electronics, mechatronics, communications technology, programming, etc.) as well as the ability to adopt the knowledge from different disciplines to synthesize and thus achieve the desired result.

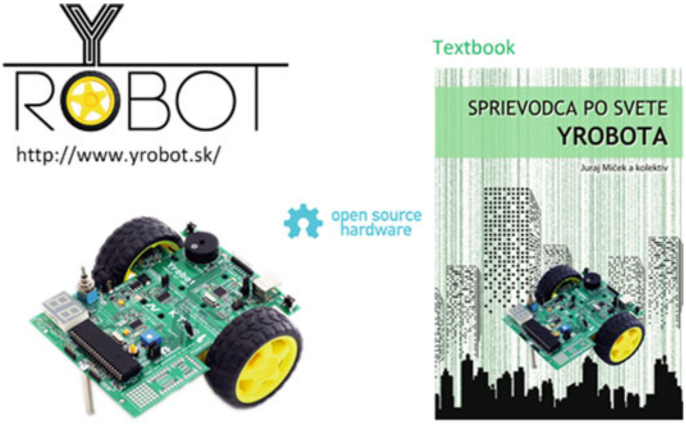


Fig. 1 Yrobot with textbook (Slovak Release)

Another result of such complex challenges treatment is the fact that solving them asks for the creation of interdisciplinary teams. This represents a natural way to support and develop the ability of students to work in teams:

- to respect the opinion of colleagues;
- to find a way and courage how to defend the ideas;
- to develop interesting approaches of colleagues
- to find effective solutions; etc.

Relatively high expectations of the Yrobot-project-team were put in the ability of students to process and develop inventing tasks by themselves. In this area, the great inventiveness of students was expected. We assumed that we meet original ideas and interesting solutions that would not be burdened with conventional approaches. However, during the first year of Yrobot usage, we saw relatively small activities of students to develop new extension modules and related new applications (e.g. [15]).

This can be on the one hand caused by generally poor knowledge of students of technological processes necessary for the HW production. On the other hand, financial demands play also an important role. For these reasons, we have decided to support the development of new applications by using of special extension modules that extend application possibilities of the basic kit.

When designing such modules, it is necessary to respect existing communication capabilities of the Yrobot base (UART/SPI) as well as mechanical and space restrictions. From the viewpoint of price minimizing, it is necessary to minimize PCB dimensions and implement a kind of simple circuit solutions while keeping an adequate reliability and flexibility.

Table 1 Communication technologies

Technology (2.4 GHz)	Standard	Module	Max bit rate
WiFi	802.11 b/g/n	WizFi250	65 Mbps
Zigbee	802.15.4	JN5168M0 MRF24J40	1 Mbps 250 kbs
Bluetooth v.2.0 EDR	802.15.1	BTM182	3 Mbps

2 RF Communication Modules

According to the evaluation of study programs of the Slovak high schools, taking into account current trends in IT development, the field of wireless communication was chosen as one of the popular application areas for Yrobot expansion. It can be assumed that additional wireless connectivity added to mobile robots extends significantly the potential application cases and scenarios.

The Yrobot was originally developed as an autonomous Yrobot device capable to solve simple tasks by reading the status info of installed sensors (e.g. moving across the line, avoiding obstacles, discovering the space, etc.).

Implementation of wireless communication allows transformation of Yrobot from single and autonomous functioning to robust multi robotic system able to solve the robust challenges and to bring the complex solutions. For an effective operation of the system it is possible to use various communication technologies, protocols and different network topologies. In our approach, we decided to implement three separate communication modules operating in the 2.4 GHz ISM band. The chosen standards can be seen in the Table 1.

All developed modules, with dimensions of 42.5 × 60 mm, can be connected via connectors on the motherboard to the Yrobot MCU. Through connector JP2 is assured power feeding of the module and the serial asynchronous communication with Yrobot MCU. Synchronous communication interface SPI employs JP9 connector. The module contains circuitry to secure the power supply, to convert the signal logic levels, and to select the communication line between UART and SPI. Module states can be controlled through signalization LEDs. The buttons on the module allows basic control or setting up of the operation mode.

3 Communication Module Y-WiFi

As a first module from the mentioned RF Communication Modules was realized communication module Y-WiFi allowing to manage and monitor the performance of Yrobot systems by using of WiFi network (IEEE 802.11x). The module should primarily secure the wireless communication between mobile Yrobots.

As the basic component of this module was chosen communication module WizFi250. WizFi250 comprises an integrated LNA as well as integrated antenna

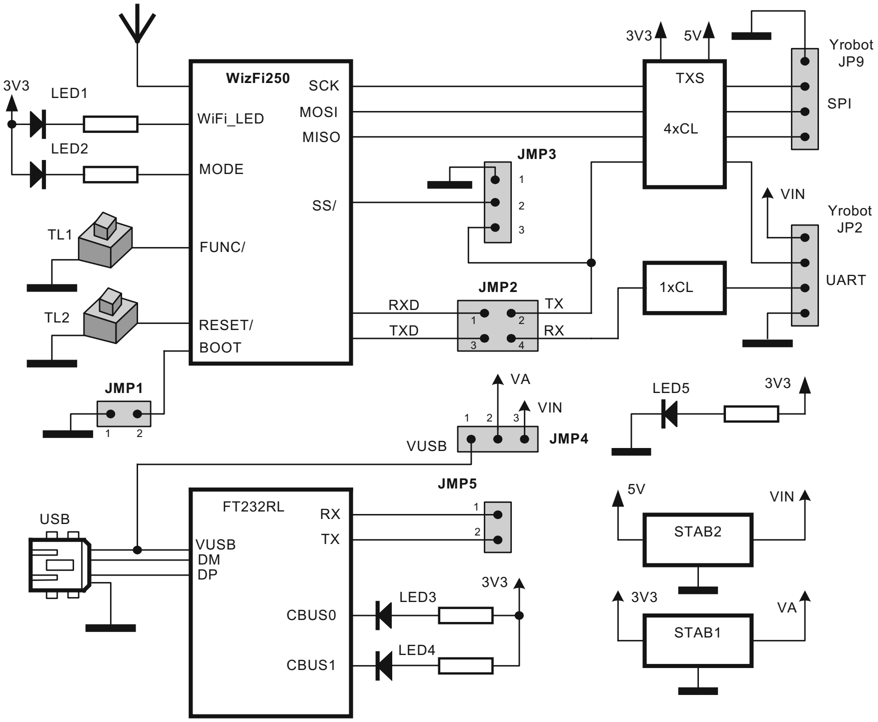


Fig. 2 Block schematics of Y-WiFi module

with connector for an external antenna connection. It can serve the secure communication protocols WEP, WPA, WPA2-PSK. Its control can be realized through a suitably chosen set of AT commands.

Module Y-WiFi integrates all other necessary components making it easy to interconnect the module with Yrobot or with personal computer via USB port. The conversion between UART and USB was assured through FT323RL circuit. Block diagram of the Y-WiFi module is shown in the Fig. 2. Yrobot with attached Y-WiFi module is shown in the Fig. 3.

3.1 Operation Modes of Y-WiFi Module

Communication module Y-WiFi can work in the two basic configurations/modes (mode Yrobot and mode PC) and in one auxiliary configuration in which it is not used WizFi250 module but Yrobot can communicate through USB port with master system.

1. **Yrobot Mode**—module Y-WiFi is connected via JP9 and JP2 to the Yrobot mother board. In this mode, the Y-WiFi module enables Yrobot to communicate

through wireless WiFi network. In the basic mode it is possible for the communication between the Yrobot MCU and Y-WiFi module to use SPI or UART interface.

   (a) In case that SPI communication has to be used, it is possible to apply TX signal as the device slave select signal (SS). Then it is necessary to interconnect pins 2–3 on the jumper JMP3, while on the jumper JMP2 it is necessary to disconnect the pins 1−2.
   (b) In case an asynchronous serial communication UART has to be used, it is necessary to interconnect pins 1–2 and 3−4 on the jumper JMP2 and on the jumper JMP3 disconnect the pins 1–2.

2. **PC Mode—**module Y-WiFi is through a USB cable connected to a host system (PC, tablet, etc.). Personal computer identifies the connection of a new device and automatically creates a virtual COM port. Then it is possible to communicate with the module via an application that enables serial communication (e.g. Hyperterminal, …). The module is powered from the USB port. In this case, it is necessary:

   (a) to disconnect the pins 2−3 on the jumper JMP3;
   (b) to interconnect the pins 1/JMP5 with 1/JMP2;
   (c) to interconnect the pins 2/JMP5 with 2/JMP2;
   (d) to interconnect the pins 1−2 on the jumper JMP4.

After the setting up of all module jumpers (JMPx), it is possible—via the master system—to communicate with the module and to control its activity by using of AT commands. Eventually it is also possible to create and verify additional WiFi-based communication applications. The additional modules (SW or HW), which can be connected through PC virtual COM port, allow the development of other network applications—from simple peer-to-peer connections, through private WiFi networks up to the complex client/server applications.

In this mode, it is also able to update the firmware of WizFi250 module. When updating, it is necessary to interconnect pins 1−2 on the jumper JMP1. When these pins are connected, signal BOOT gets to the LOW state. This causes WizFi250 system transition into the BOOTING mode after the system restart (Tl2 button).

3. **Auxiliary Mode**—Y-WiFi module configured in this mode utilizes the circuitry that secures conversion UART/USB (FT232). According to this, it is able for the Yrobot MCU to communicate with the superior master system (PC, tablet, etc.). This makes it possible to control Yrobot platform through the commands entered from the connected computer. In this mode it is necessary to interconnect pin 1 on the jumper JMP5 together with the pin 2 on the jumper JMP2as well as the pin 2 on the jumper JMP5 with the pin 4 on the jumper JMP2. It is

also recommended, in auxiliary mode, to power the Y-WiFi module from the Yrobot motherboard. Therefore, it is necessary to interconnect the pins 2–3 on the jumper JMP4.

4 RF Communication Modules Y-Bluetooth and Y-ZigBee

Communication modules Y-Bluetooth (Y-BT) andY-ZigBee have the similar characteristics as the previous module Y-WiFi. When Bluetooth connectivity has to be implemented, BT modules BTM11x or BTM18x could be used instead of WizFi250. When the ZigBee communication standard is selected, modules MRF24J40x or JN5168x could be used.

The selected Bluetooth module has to be assembled on the Y-WiFi PCB through the component side. Communication with Yrobot MCU can be realized through UART or SPI interface. The module can be controlled by a set of AT commands as well. Y-Bluetooth supports PC and Yrobot Modes.

The same assembly procedure as for the Y-BT is valid for the Y-ZigBee too.If the user chooses the Y-ZigBee solution, the ZigBee module should be placed from the component side of PCB. From the supported Zigbee modules, (MRF24J40MA/MB, or JN5168), only the JN5168 supports UART/SPI communication, while MRF24J40 can communicate with Yrobot MCU only via SPI.

Fig. 3 Yrobot with Y-WiFi module

5 Conclusion

An extension of the Yrobot kit by the set of network modules significantly expands the variety of applications that can be implemented under it. The kit is since its inception conceived and designed for the needs of teaching of subjects in IT. In addition to this primary function, the kit serves also the popularization function. The kit should be used for an encouragement of the students for the study of the technical subjects/fields.

In the near future, the focus will be put on the development of interesting and original applications designed according to experiences with the communication modules usage. The delivery of supplementary textbook is in this case more than necessary. In the textbook, the basic capabilities of individual RF network technologies supplied with the simple examples, that will illustrate the benefits and limitations of wireless communication, will be described. Hopefully, other interesting applications, which could motivate the students to the own further development, will be part of the textbook too.

Further steps are, beside the textbook development, oriented in the development of additional modules in the field of RF communication. At the present time, RFID, NFC, Z-Wave modules, together with the chosen proprietary communication systems in the free ISM bands (e.g. RFM70), are being developed. It is expected that these extensions will expand the current status of the kit with other interesting ICT applications.

References

1. Gomez-de-gabriel, J.M, Mandow, A.., Fernandez-lozano, J., Garcia-cerezo, A.: Mobile robot lab project to introduce engineering students to fault diagnosis in mechatronic systems. IEEE Trans. Educ. **58**(3), 187–193 (2015). doi:10.1109/TE.2014.2358551
2. Cubero, S.N.: A fun and effective self-learning approach to teaching microcontrollers and mobile robotics. Int. J. Electr. Eng. Educ. **52**(4), 298–319 (2015). doi:10.1177/0020720915585798
3. Lopez-rodrigues, F.M., Cuesta, F.: Andruino-A1: low-cost educational mobile robot based on Android and Arduino. J. Intell. Robot. Syst. **81**(1), 63–76 (2016). Special Issue: SI, doi:10.1007/s10846-015-0227-x
4. Rubenstein, M., Cimino, B., Nagpal, R., Werfel, J.: AERobot: an affordable one-robot-per-student system for early robotics education. In: IEEE International Conference on Robotics and Automation (ICRA 2015), 26–30 May 2015, pp. 6107−6113, Seattle, WA, USA. ISSN:1050-4729, doi:10.1109/ICRA.2015.7140056
5. Suzuki, K., Kanoh, M.: Effectiveness of a robot for supporting expression education. In: Conference on Technologies and Applications of Artificial Intelligence (TAAI 2015), 20–22 Nov 2015, pp. 498−501. doi:10.1109/TAAI.2015.7407119
6. Zhi, D., Wu, Z. Li, W., Zhao, J., Mao, X., Li, M., Ma, M.: Education-oriented portable brain-controlled robot system. In: IEEE International Conference on Robotics and Biomimetics (2015 ROBIO), 6–9 Dec 2015, pp. 1018−1023, Zhuhai. doi:10.1109/ROBIO.2015.7418905

7. Fabregas, E., Farias, G., Dormido-canto, S, Guinaldo, M, Sanchez, J., Bencomo, S.D.: Platform for teaching mobile robotics. J. Intell. Robot. Syst. **81**(1), 131–143 (2016). doi:10.1007/s10846-015-0229-8
8. Binugroho, E.H., Pratama, D., Rizqy Syahputra, A.Z., Pramadihanto, D.: Control for balancing line follower robot using discrete cascaded PID algorithm on ADROIT V1 education robot. In: International Electronics Symposium IES 2015, 29–30 Sept 2015, pp. 245−250, ISBN: 978-1-4673-9344-7, Surabaya, Indonesia. doi:10.1109/ELECSYM.2015.7380849
9. Kettler, A., Szymanski, M., Liedke, J., Wörn, H.: Introducing Wanda—A new robot for research, education, and arts. In: International Conference on Intelligent Robots and Systems (IROS 2010), IEEE, 18–22 Oct 2010, pp. 4181−4186, ISSN:2153-0858, Print ISBN 978-1-4244-6674-0, Taipei. doi:10.1109/IROS.2010.5649564
10. Torres-torriti, M.: Arredondo, T, Castillo-pizarro, P: Survey and comparative study of free simulation software for mobile robots. Robotica J. **34**(4), 791–822 (2016)
11. Banduka, M.L.: Robotics first-a mobile environment for robotics education. Int. J. Eng. Educ. **32**(2), 818–829. Part A
12. Kochláň, M., Hodoň, M.: Open hardware modular educational robotic platform—Yrobot. In: RAAD 2014, 23rd International Conference on Robotics in Alpe-Adria-Danube Region, Slovakia (2014). ISBN 978–80-227-4219-1
13. Miček J., Karpiš O.: Audio communication subsystem of multi-robotic system YROBOT. In: RAAD 2014, 23rd International Conference on Robotics in Alpe-Adria-Danube Region, Slovakia (2014). ISBN 978-80-227-4219-1
14. Miček, J. et al.: Sprievodca po svete Yrobota, Žilinskáuniverzita (2015). ISBN 978-80-554-1120-0
15. Miček J., Karpiš, O., Kochláň M.: Audio-communication subsystem module for Yrobot—a modular educational robotic platform. In: EDERC 2014, 6th European Embedded Design in Education and Research, Italy (2014). ISBN 978-1-4799-6842-8

Aeris—Robots Laboratory with Dynamic Environment

Michal Chovanec, Lukáš Čechovič and Lukáš Mandák

Abstract Is it possible to create an ant robot which can leave "scents" behind? Is it possible to create robotic football on the same platform? What about mouse maze, line follower or sumo robot? And what about the possibility to have dynamically scalable environment which can interact with robots? Everything mentioned and more can be done on the same platform called Aeris. In this paper, robot platform called Aeris is presented. Department of Technical Cybernetics has long years experience with implementing different robots into education. The logic step was trying to integrate existing robots experiment scenarios into one platform but the result went over borders of common "robot" discipline. Universal and interactive robot "playground" concept is presented, with potential to be easily usable on teaching purposes from simplest robotics to technical cybernetics, embedded systems or artificial intelligence. This platform has a potential to be also powerful equipment for researchers, for example in dynamic learning systems, swarm systems or other learning algorithms. The actual state of Aeris is presented with an overview of future work.

Keywords Robots in education · Robot laboratory · Dynamic map on LCD · Robot simulations

1 Introduction

It is said that if we want to see best of ourself we should let ourself play.

1.1 History of Robots on DTC

Department of Technical Cybernetics implements robots into education since 2002 when AT90S1200 [1] microcontroller was released. Our first two wheel drive robot

---

M. Chovanec · L. Čechovič (✉) · L. Mandák
Faculty of Management Science and Informatics, Department of Technical Cybernetics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovakia
e-mail: lukas.cechovic@fri.uniza.sk

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_16

with stepper motors and ultrasound sensors was built on this device. As this was introduced into education, the interest from the students side was enormous. Next step was to bring some challenges into the process. ISTROBOT [2] competition on Slovak Technical University became an arena for the students to measure technical and software parts of their robots. Every year since then, students were guided to build their own or use existing wheel drive robots with various microcontrollers and sensors prepared specially for competition categories as line follower, mouse maze or sumo [2].

These activities were focused on University students and their study process. Figure 1 shows presentation of robot called George.

Then there come interest from high schools for education platform and as there was importance for better preparation for potential students, platform YROBOT was created in 2012 [5, 6]. Parallel with developing YROBOT, faculty became the host of First Lego League [4] regional competition from 2012. This competition is primarily focused on primary and secondary school students.

Since then DTC helps to bring the whole scale of students at different school levels to robotic education.

In late 2014 research need brings idea of an ant robot which can leave mark—scent behind, as a part of an ant colony optimization problem. Standard robot approach with the static track was not sufficient so it become clear that there is a need to bring a kind of interaction upgrade to the existing approach. This effort resulted into the system called Aeris.

Fig. 1 Presentation of robot called George on European Researchers' Night in Bratislava, Slovakia, 2011. It was the predecessor of YROBOT

1.2 *Standard Robot Playground*

In standard robot scenario there is defined static playground (track). Usually, this track is for one robot goal, painted on the floor, created with tape, build from the wooden board or Lego parts. Doing changes into track needs to manually reassemble track parts.

1.3 *Dynamic Robot Playground*

In dynamic robot playground, the map is changeable at any moment. This fact brings many new possibilities and applicability of such a system. This system can handle all standard static robot scenario maps as line follower, sumo or mouse maze. It is possible to extend these standard scenarios for example in a way such as a line follower track could dynamically change while the robot is following, or maze could have dynamic parts. It is easy to imagine many new parameters in standard static robot scenarios.

1.4 *Related Work*

As idea of using robots for teaching is not new so let's highlight some interesting systems incorporating robots into education and research.

SyRoTek project that aims to create an e-learning platform for supporting teaching mobile robotics and artificial intelligence [7]. In principle final version of Aeris should be able to work also as remote e-learning platform thanks to its client-server based programming model with high modularity.

Robot platform Colias. "Colias is a low-cost, open-platform, autonomous micro robot which has been developed for swarm robotic applications. Colias employs a circular platform with a diameter of 4 cm. It has a maximum speed of 35 cm/s that gives the ability to be used in swarm scenarios very quickly in large arenas. Long-range infrared modules with adjustable output power allow the robot to communicate with its direct neighbors from a range of 0.5 cm to 3m″ [8]. If Aeris system is a grey-scale system Colias could be a possible choice of the robot to work with. Missing are RGB sensors, WIFI module, and computation power, so instead of making changes into Colias, own robot platform was developed.

The main advance in Aeris robot is more computing power with 72MHz ARM CortexM4 with FPU on board and thanks to used API, robot programming stayed user-friendly. This allows computing more complicated problems from artificial intelligence. Such neural networks, reinforcement learning or genetics algorithms, all of them require adequate memory and computing power to be processed in real time.

Cosϕ. Cosϕ is artificial pheromone system that shows similarities with Aeris as it uses also LCD screen and USB camera [9]. Cosϕ system is specialized on ant colony optimization using Colias robot and is basically one possible grey-scale scenario of Aeris system. Differences of Aeris system compare to Cosϕ are that Aeris is aiming to be the universal system, with possible to create unlimited numbers of scenarios and Aeris is recognizing RGB components in 10-bit resolution.

2 Aeris System

Aeris is aiming to be the universal tool for creating dynamic 2D environment simulations able to interact with mechanical robots. Created environment itself does not need mechanical robots as it is able to create virtual robots too because every part of the system is the robot (as example wall is a non-movable robot with specific diameters). Part of Aeris responsible for simulations and interaction with mechanical robots is Supervising Control and Simulating center. Trough this part of system, map and robots are able of interaction. System control center knows position of the mechanical robot (thank's to camera above playground or touchscreen) and is able to communicate with the robot. Control center controls scenario which is displayed on the map. The map is represented by horizontally placed LCD display. The robot is reading a map and executing given tasks by given program.

2.1 Aeris Parts

Aeris, as is defined, needs playground/environment visualization device represented by horizontally placed LCD display and control center represented by a computer which runs program and control display. To connect Aeris with real mechanical robots, control center needs to be equipped with camera or touchscreen. Real robots, to work with system, needs to be able to read color information from display and communicate with control center or communicate together in multiple robot tasks. Basic functional diagram of the system is showed in Fig. 2.

Basic parameters of robot are RGB sensors to read LCD display and Wi-Fi device to communicate with the system. For the system purpose, universal version of the robot is aiming to handle as much as possible scenarios for the user, with space to extend robot parameters for specialized tasks.

- Universal robot
  - RGB sensors
  - 3D gyroscope
  - 3D accelerometer
  - 3D compass
  - Microcontroller
  - Drives with controllers

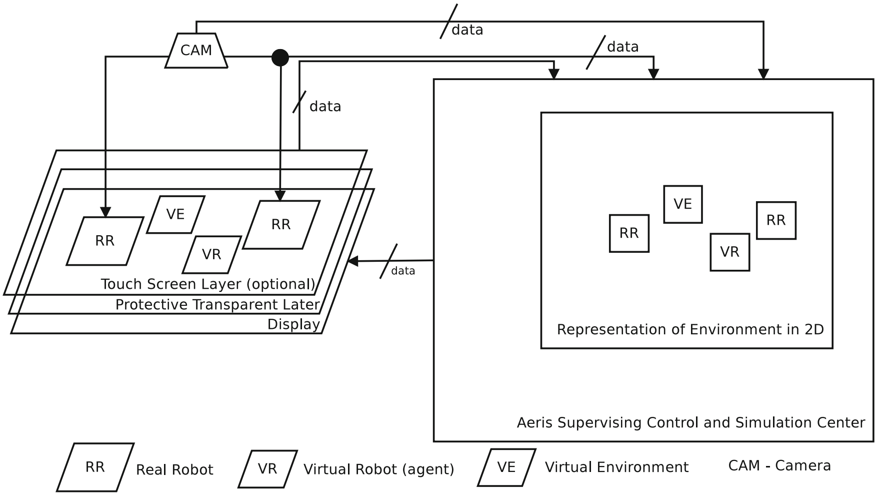


Fig. 2 Aeris simplified system functional diagram

- Wi-Fi communication module
- Slots for expansion boards

Visualization device is horizontally placed LCD display, protected with polymethylmethacrylate 2 mm thick transparent layer. As an extension, there is a goal to add touchscreen layer to extend possibilities to track robots on display.

- Playground physically created with horizontally placed LCD display.
  - LCD Display
  - Protective layer
  - Touch layer (optional)

Control center is a program with client-server based programming model with high modularity running on a computer, controlling system tasks as LCD driving, camera reading, Wi-Fi communication, environment simulation, and algorithms processing.

- Supervising Control and Simulating Center tasks
  - LCD display controlling (graphical input to LCD panel)
  - LCD display External state reading (Camera reading the activity of physical robots on the map)
  - Mechanical robot communication (Wi-Fi module)
  - Environment processing
    Virtual robots processing
    Real robots processing
    Virtual environment processing
    Real environment processing
    Algorithms

2.2 Platform for New Dynamic Playground—RGB Sensors and LCD Display

The basic goal for the transition from a static playground to the dynamic playground was to find the platform for dynamic playground which can be readable with sensors in a similar way as standard static robot playground. For this purpose, LCD LED display was used. In a horizontal position, it is "easy to get" equipment with sufficient dynamic parameters dependent on display type (frame rate).

Next step was to find and test sensors, applicable on the robot, able to read information from common LCD displays. RGB sensor APDS9950 was chosen and tested as it is Digital Proximity, RGB, and Ambient Light Sensor. This device is capable of up to 390 readings per second. It is characterized by easily applicable I2C communication interface. Practical tests showed that it is able to read surface of LCD display from needed range, approximately 1.5 cm in our scenario.

At the beginning, obtained values showed noise in intensity of color components, Fig. 3, which was partly degrading information from sensors. This did not prevent the color identification as the ratio between color channels stayed in sufficient range to distinguish basic colors.

It was found out that sensors were affected by noise in an intensity of color components while displayed brightness value was under 100 %. With display brightness level on 100 %, intensity noise disappeared (Fig. 4). It is supposed to be the result of display backlight controlling.

Next thing to deal with was a frame rate of the display. To minimize this possible effect, sensors were set to 250 readings per second, which was over 4 times higher than 60 Hz frame rate of used display. In measuring, as is seen in Fig. 4, random spikes in color components were present, which is supposed to be frame rate artifacts. These were filtered out with a median filter.

- Parameters of Aeris dynamic playground
  - LCD LED Display Dell E5515H [3]
    Diagonal Viewing Size 54.6" (1386.84 mm)
    Resolution 1920 × 1080 pixels
    60 Hz Frame Rate
  - RGB sensor APDS 9950
    Up to 390 readings of color per second
    I2C communication interface
    Small Package Dimensions (3.94 × 2.36 × 1.35 mm)

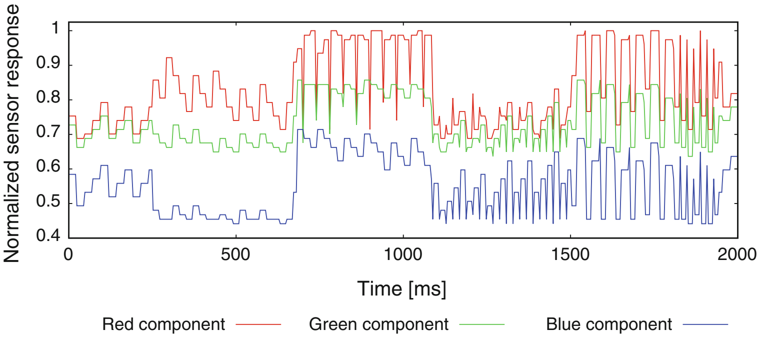


Fig. 3 Sensors response on *white light* with LCD LED brightness setting on 50 %, 250 Hz sensor sampling

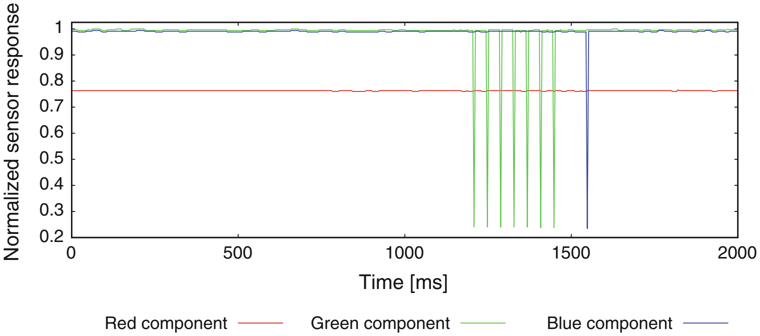


Fig. 4 Sensors response on *white light* with LCD LED brightness setting on 100 %, 250 Hz sensor sampling

2.3 Aeris Supervising Control and Simulation Center

In proposed solution, it was necessary to split system parts into independent blocks to maximize universality and variance of experiments.

- environment
- map
- server
- robots
- visualization

Main part of the system is running environment, which is handling incoming agents (robots) requests Fig. 5. Environment is processing all interactions including physics, communication, and rewards for agents. Environment is loading map, which can be designed manually or randomly (to simulate real world). All these parts are connected using common TCP server, which has open connections for robots and

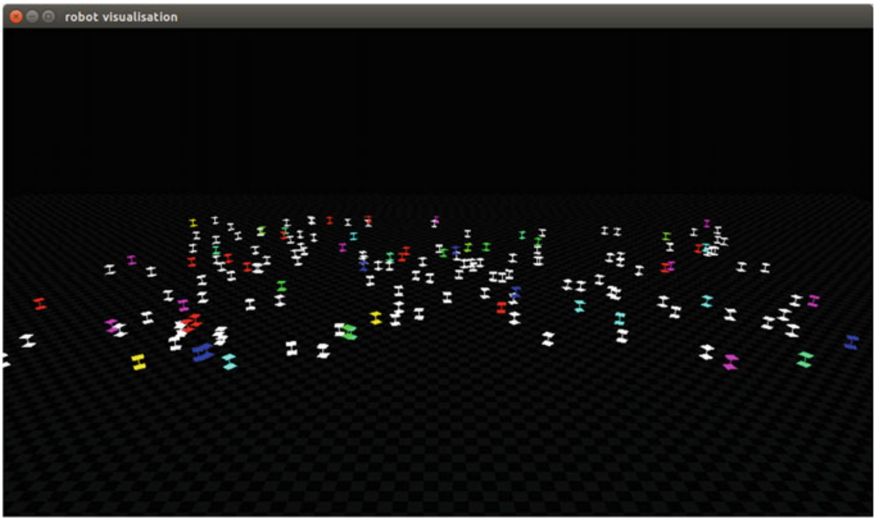


Fig. 5 Visualization of virtual robots in Aeris system in 3D

visualization. Each robot can be calculated on its own computer as this is necessary for complex AI algorithms like huge neural networks real-time calculations. More than one visualization computer can be also used, so the experiment can be visualized on any machine connected to the server via Internet. It is planned to do visualization on html5 currently, so no other application will be necessary.

2.4 Possibilities and Limitations

Aeris system is able to handle static and dynamic scenarios. The size of scenario is limited by the used display. Dynamic scenarios are limited by the frame rate of used display and reading rate of used sensors for display reading.

From its nature, the system is able to create 2-dimensional scenarios or 2-dimensional version of specific 3-dimensional scenarios. As example, in mouse maze it is possible to represent near presence of wall by increasing intensity of color near the wall, to use a camera on robot pointed on the surface or use system camera and give the information to the robot that the wall is near.

Aeris Supervising Control and Simulation Centre is being constantly developed to become map editor, map generator, center for handling various simulations of virtual robots behavior or maps behavior center to create "living" maps.

Fig. 6 Aeris robot version 1 as line follower

2.5 *Actual State and Future Work*

The actual state of Aeris system is far from being the final version.

The control program is able to create and simulate robots. The simple static map editor was created. Static and dynamic map for line follower was created Figs. 6 and 7. The universal mechanical robot was created and tested. All development is under Linux in C and C++.

Actual work is focusing on mouse maze scenario, visual communication LCD to robot, camera localization of robot with help from students working on their thesis but basically, all parts of the system are being constantly upgraded. In next step, there is a goal to bring user-friendly version easily usable for teaching simple robotics on schools.

- Actual state and work in progress
  - Robot version 2—done Figs. 8 and 9
  - Robot programs for specific scenarios—done/development
  - Supervising Control and Simulation Centre—done/development

Fig. 7 Early version of Aeris system

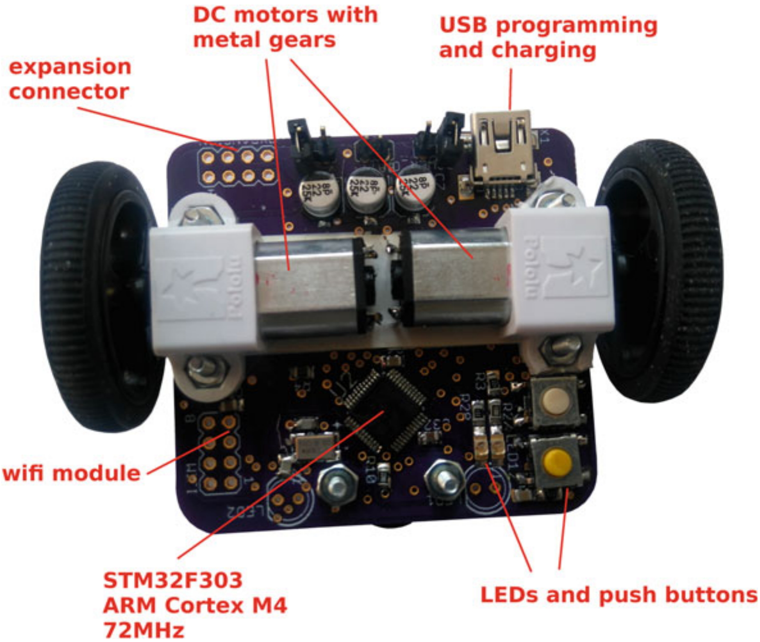


Fig. 8 Aeris robot version 2, *top view*, 5 × 5 cm base dimensions without wheels

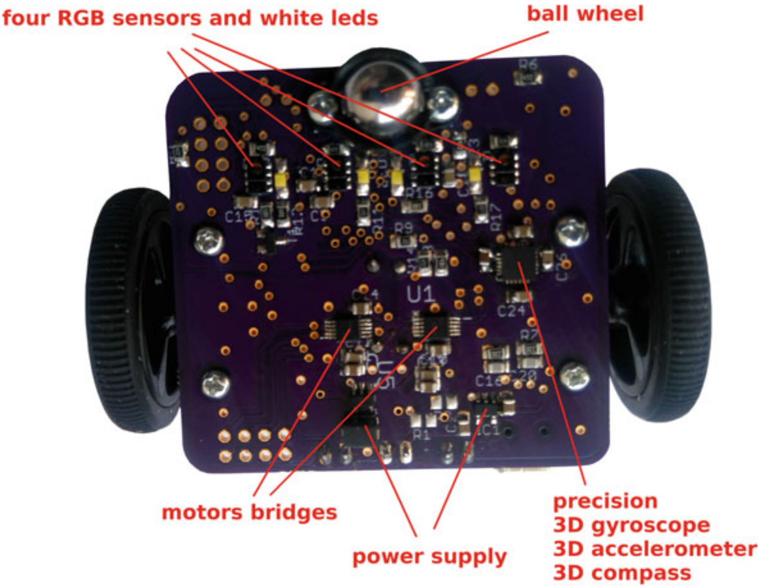


Fig. 9 Aeris robot version 2, *bottom view*, 5 × 5 cm base dimensions without wheels

  - Camera detection of robot position—development
  - Optical monitoring of robot communication—development
  - Optical robot-to-camera communication—development
  - Expansion board to enhance robot functions—development

- Future work
  - Charging stations for robots
  - Touch screen addition to the system
  - Version for teaching simple robotics on schools
  - Create robotic competition for Aeris system

Recently, the second version of a robot with diameters of base 5 × 5 cm was created Figs. 8 and 9. This robot replaced first unreliable but smaller version with base 3 × 3 cm. Aeris robot was created with the aim on universality, where the goal was that robot will be able to handle all tasks what the system could bring.

Project web page is being developed including information about the project [10].

2.6 Aeris System in Teaching Process

Aeris system is already incorporated into teaching process on Computer engineering study program on Faculty of management science and informatics in subjects Project 1 and 2, and also, there are preparations for another subject. Also partial works on the project are doing troughs bachelor's and master degree thesis. Aeris is being used on presentation activities for public with the aim to promote the study of science and engineering. Co-author of this paper is master degree student. The process of creating Aeris is itself incorporated into the teaching process.

The final version of the system could handle most of the present days task needed for teaching robotics as simplest presentation scenarios, robotic disciplines scenarios or support for the Ph.D. study of Artificial intelligence and computer engineering.

3 Conclusion

Aeris system is aiming to become powerful robotic laboratory equipment by its unprecedented variability, ability to change on fly different scenario maps and robots behavior programs. With Aeris, there is a possibility to bring till now simulation only experiments closer to the real world, closer to audience. Evacuation problems, multi-agent problems, robotic challenge scenarios with totally new possibilities to create interactive maps by a new and fresh way, all together on the same platform. Real robot interacting with simulated robots on "living map". Implementing simplest robot scenarios as line follower, trough robotic football to ant colony. Limits of playing with robots are pushed forward with Aeris system.

Acknowledgments This work was supported by Foundation of Kia Motors Slovakia in Foundation Pontis in project Umelá inteligencia hravou formou.

References

1. 8-bit AVR Microcontroller with 1K Byte of In-System Programmable Flash (2002). http://www.atmel.com/images/doc0838.pdf
2. Balogh, F.: Mobile robots contest ISTROBOT. Acta Mechanica Slovaca, 2-A/2006 (2006)
3. Avago Technologies: APDS-9950 Digital Proximity, RGB and Ambient Light Sensor (2015). http://www.avagotech.com/docs/AV02-3959EN
4. First Lego League. http://www.firstlegoleague.org/
5. Miček, J., Karpiš, O., Kochláň, M.: Audio-communication subsystem module for Yrobot—a modular educational robotic platform. In: EDERC 2014 [electronic source]: Proceedings of the 6th European Embedded Design in Education and Research: 11–12 Sept., 2014, Milano-Italy.-[S.l.]: IEEE, 2014.-978-1-4799-6842-8.-CD-ROM, p. 60–64 (2014)
6. Kochláň, M., Hodoň, M.: Open hardware modular educational robotic platform—Yrobot In: RAAD 2014 [electronic source]: 23rd International Conference on Robotics in Alpe-Adria-Danube Region: 3–5 Sept. 2014 Smolenice Castle, Slovakia: conf. proc. - Bratislava: Slovak University of Technology, 978-80-227-4219-1 (2014)
7. Kulich, M., Chudoba, J., Kosnar, K., Krajnik, T., Faigl, J., Preucil, L.: SyRoTek–distance teaching of mobile robotics. IEEE Trans. Educ. **56**(1), 18–23 (2013)
8. Arvin, F., Murray, J., Zhang, C., Yue, S.: Colias: an autonomous micro robot for swarm robotic applications. Int. J. Adv. Robot. Syst. **11**(113), 1–10 (2014)
9. Arvin, F., Krajník, T., Turgut, A.E., Yue, S.: COSϕ: Artificial pheromone system for robotic swarms research. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 407–412. Hamburg (2015)
10. http://aeris.fribot.sk/

`UNC++Duino`: A Kit for Learning to Program Robots in Python and C++ Starting from Blocks

Luciana Benotti, Marcos J. Gómez and Cecilia Martínez

Abstract We present `UNC++Duino`, an open source educative software for learning to program a robotic kit in C++ and Python. Besides of these two industry programming languages, `UNC++Duino` can be programmed using 2 high level languages based on blocks are free of syntax errors. One of the block based languages included is completely iconic allowing for its use with preliterate children. The hardware we use with `UNC++Duino`, the open RobotGroup robotic kit, can be used to build different automated constructions based on an Arduino board, sensors and actuators. `UNC++Duino` was developed within Argentinean K-12 schools by the Universidad Nacional de Córdoba with the collaboration and support of the Argentinean National Ministry of Science and the RISE program in Google for Education. Its goal is to provide an engaging tool for learning to program in different programming languages with increasing difficulty and control of the hardware.

1 Introduction

Why is it necessary to find effective and innovative ways of engaging more students into Computer Science (CS)? Taking our country as an example, we know that Argentinean universities graduate approximately 4000 CS students a year (compared to 10,000 in Law and 15,000 in Economics) while the national industry needs to hire twice that amount per year. The lack of human resources in CS has an economical impact in Argentina. The Information and Communication Technology (ICT) industry in Argentina, despite having grown intensely in the last ten years (four times in

---

L. Benotti (✉) · M.J. Gómez · C. Martínez
Universidad Nacional de Córdoba, Haya de la Torre y Medina Allende,
Córdoba, Argentina
e-mail: benotti@famaf.unc.edu.ar; luciana.benotti@gmail.com
URL: http://masmas.unc.edu.ar

M.J. Gómez
e-mail: mgomez4@famaf.unc.edu.ar

C. Martínez
e-mail: cecimart@gmail.com

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_17

the number of employees, and nine times in the amount of exports), is still struggling to find qualified workers for its workforce.

Part of the reason why students do not choose to pursue a career in CS is that, neither programming, nor other CS techniques and concepts, are taught at school. Previous studies show that this lack of early CS education can influence career choices; students may not be selecting CS simply because they do not know what CS is [4, 10]. Since it is not taught at school, misconceptions about what the discipline actually is are commonplace. Indeed, recent work found that although more than 90 % of Argentinean high school students surveyed use computers, most of them believe that programming means installing programs [2].

The typical K-12 student in Argentina never encounters CS topics during his/her school years. Programming is never taught at school, not even as an optional course. The ICT curriculum in Argentina focuses on user training classes rather than CS content. In these courses, computing entails little more than learning how to use a word processor, a spreadsheet or how to write an online blog. Students often get bored in their ICT classes and outperform their own teachers. This context is not unique to Argentina; many developed countries share the same problem. For example see the cases of the USA [20] and the UK [9]. There are some exceptions such as Israel, where CS has been taught at (some) high schools for many years now [22], and other countries are starting to follow. This is the case, for example, of New Zealand [1].

Children are using computers much earlier each decade. However, this intensive use is not contributing to their knowledge of CS as a discipline nor to developing the ability to understand a programming language. Children become software consumers very early but they do not learn the basics of how this technology works.

Given the current situation, there is increasing consensus that teaching programming in particular, and CS in general, in an engaging way in the school curriculum is necessary. It is imperative to both include students in the technological world as active and creative citizens, and to help them make educated choices about their professional future.

Bergen [3] points out that programmable toys such as robots, give children the possibility of creating, imagining, and exploring. Robot programming is at the same time engaging, stimulating and rich of many important CS concepts where the digital world connects to the real world.

With the goal of teaching programming and other CS important concepts in an engaging and interdisciplinary way the main contributions of this paper are:

- Introducing a multilanguage robot programming platform that allows children to change from one language to another making evident the unimportance of the particular syntax, helping students discover new concepts and learning new techniques when they are ready.
- Describing an open source educative robotic kit that can be used to build an unlimited number of automated prototypes based on Arduino boards using just a screwdriver—a 3-wheel robot, an elevator, a harvester, an automated house, etc.
- Proposing a set of original activities and suggestions for adapting the same activity for preliterate, primary school and highschool students. The activities make use

of the programming kit to teach not only programming but also propose how to integrate content from other disciplines (e.g. astronomy and physics).

2 Previous Work

Starting very young, children can create, run and debug simple computer programs using programming languages that are both challenging and attainable for their age [5, 14]. Even children who are preliterate, have diverse background and different developmental stages, can program tangible platforms using iconic commands [7, 11]. The effects and implications of learning CS at such an early age have been analyzed in previous work. According to Clements [5], young children who program concrete objects have the opportunity to analyze a situation and reflect on the properties of the objects they have to manipulate.

While exploring how to teach CS to little children, researchers found that the difficulties in children programming laid in their immature motor skills and on syntax problems [8, 16]. Thus, there has been an important development on specific programming platforms to address the developmental traits of preschool children (such as Toon Talk [17], Scratch Jr [15], Cherps [19], among others). The kit we present in this paper is one of such programming platforms, particularly designed for programming robots and other automated constructions. Differently from previous work, it offers an integrated way to grow from simple block programming languages into full fledged languages such as Python and C++.

Flannery and Berns [7] showed that as a result of robot programming in preschool, children imagine, plan its action, and construct a robot. In their study, the authors found that all 4–6 year old students could program short challenges and explore robot's capabilities.

Although most interventions to teach programming with robots achieve high student engagement and task completion, most available robotic programming kits are not accessible to schools for their high cost or their lack of flexibility. Some kits are tailored to be used with one (sometimes proprietary) block-based, or otherwise simplified, programming language. This rigidity hinders children's and teacher's creativity limiting the possibility of creating and solving new challenges appropriate for different age groups.

The following studies compare different age groups performance on similar robot programming activities. Magnenat and his colleagues [12] taught CS with robot programming to different age groups of children using an event handler language to program robot actions in response to different events. Comparing the groups performance in the same task, they found that most children understood and solved simple tasks such as moving a robot upon a touch of a button or identifying robot's instructions. However, older children performed better on complex programming that required several conditions or events. Dagiene et al. [6] compared students from Finland, Sweden and Lithuania from 7 to 12 years-old performing similar algorithmic thinking tasks exercises. Using multiple choice questions, they evaluated concepts such as algorithm modularity, data structures, and control flow. They found no strong difference across age groups, but rather among countries. The authors suggest that educational context, academic quality and in particular, reading ability promoted by each school system may be strongly related to learning programming. Thus, we want to highlight the value of developing a completely open source robotic programming kit[1] such as `UNC++Duino`. Using it, teachers and students can choose not only what kind of robotic construction they want to build—a 3-wheel robot, an elevator, a harvester, etc—, but also the programming language they want to do it in—a high level, iconic language or a low level language such as C++, or Python. Python is one of the top programming languages used nowadays by universities in introductory programming classes. Having different programming languages accessible side by side in an interactive development environment, help the students explore and move into more complex languages when they are ready.

In this paper we present the robotic kit hardware and the multilanguage programming software `UNC++Duino`. `UNC++Duino` was developed by the Universidad Nacional de Córdoba in Argentina with the collaboration and support of the National Ministry of Science and Technology and the RISE program in Google for Education. The hardware, described in Sect. 3, can be programmed in parallel in different programming languages organized in an increasing order of difficulty and control, as described in Sect. 4. Section 5 propose a set of original activities and suggestions for adapting the same activity for preliterate, primary school and highschool students. The activities make use of the programming kit to teach not only programming but also propose how to integrate content from other disciplines.

3 Hardware: The Robotic Kit

The hardware that we use was designed by the Argentinean company RobotGroup. The RobotGroup kit, when used to build a 3-wheel robot as shown in Fig. 1b, is small (13 × 13 × 12 cm) but it includes an interesting set of sensors and actuators. It has 2 engines and gearboxes of 200 rpm located in the two front wheels. It includes two IR sensors located where the "eyes" of the robot are. These sensors can be used to test for proximity. It also has two sensors on the bottom that measure the ground reflectivity and its colour. As can be seen in Fig. 1a, the Arduino board contains standard I/O connectors.

The kit also has a USB 2.0 port and 6 connectors for analogical 10-bit sensors. A programmable array of user leds is also included, as well as leds indicating power, and engine direction rotation. Extensions can be added such as the standard shields Arduino-compatibles (WiFi, Ethernet, ZigBee, extra engines, etc.). Finally, the kit also includes a microphone, a sound synthesizer and an IR sensor for remote control.

---

[1] The robotic kit software as well as the hardware are open source and the sources are available at http://masmas.unc.edu.ar and http://robotgroup.com.ar.

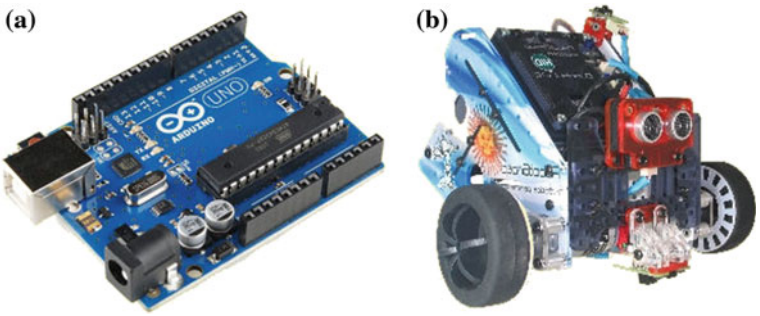


Fig. 1 The arduino board used by the kit and a 3-wheel robot built with it

We use this hardware because of the following advantages. To begin with, it is affordable and usable out of the box, allowing schools and parents to buy it with a reasonable budget. At the time of writing this paper, it was 4 times cheaper than the well known Lego® Mindstorms® (and even cheaper if less sensors and actuators of those described here are included). Moreover, the hardware design is open and based on Arduino boards supporting Grove System; which is widespread and continuously evolving. The board used is illustrated in Fig. 1a. In addition, it includes several sensors, actuators and ports in order to add extra ones. Finally, the kit is very flexible allowing for the construction of an innumerable number of automated constructions. For example, it can be used to build a 3-wheel robot (Fig. 1b), a mill or Ferris wheel (Fig. 2a), a driverless cart (Fig. 2b), a toy combine harvester (Fig. 4a), a small crane (Fig. 5b), etc.

The Ferris wheel (illustrated in Fig. 2a) can be programmed to turn on an off different leds and to play music when reacting to some programmed condition coming from its sensors. Or it can be transformed, by just using a screwdriver and some programming, in a driverless toy cart (illustrated in Fig. 2b) that can deliver toys to different rooms of a house. In the next sections we give more examples of automated constructions and how they can be programmed, but we believe that the possibilities are endless.

4 Software: Programming with Blocks, Python and C++

The `UNC++Duino` programming environment includes different programming languages. Its two block languages are based on BlocklyDuino,[2] a platform that builds on Blockly. BlocklyDuino was developed by Google for programming Arduino boards supporting Grove System[3] with blocks. A fragment of code in the block language created by us for preliterate children is shown in Fig. 3a, a fragment of the

---

[2] https://github.com/BlocklyDuino/.

[3] http://www.arduino.cc/.

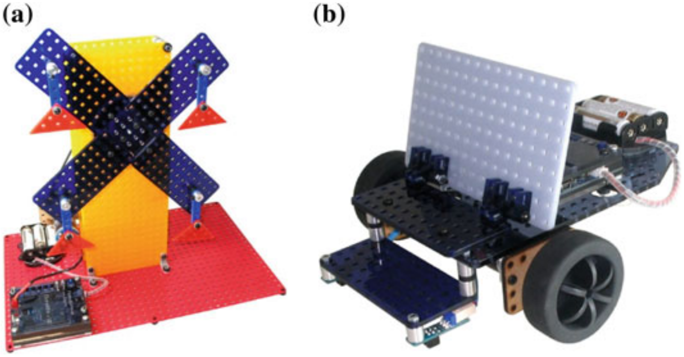


Fig. 2 An interactive Ferris wheel and a driverless cart that deliver toys

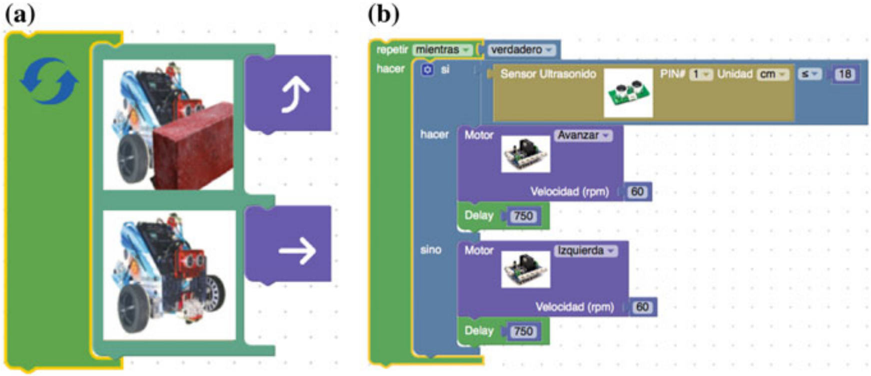


Fig. 3 The same code written in `UNC++Duino` iconic language and Blockly

original BlocklyDuino code is shown in Fig. 3b. The code has a repeat while true that includes an if-then-else instruction. This instruction means that if there is an obstacle in front of the ultrasound sensor then the robot will move forward, otherwise it will turn left.

Many educative platforms like Code.org (in their Hour of Code initiative [21]) or MIT APP Inventor [18] use Blockly. Blockly is open-source under the Apache 2.0 License. `UNC++Duino` extends BlocklyDuino to adapt it to the robotic kit described in Sect. 3 by creating new blocks for some sensors and actuators. For example, we added a block to play different songs using the robot sound synthesizer.

More importantly, we also extended BlocklyDuino to add the iconic language illustrated in Fig. 3a. `UNC++Duino` can also be programmed in other full programming languages such as Python and C++, each language was selected because it offers a different level of difficulty and expressiveness. The simplest one is the iconic language, but it is also the one that offers less control to the student. In this language the student cannot choose, for example, the speed of the robot, its speed is defined as a constant. The iconic language represents the move forward action by a straight arrow that moves the robot forward a fixed length (set arbitrarily to 20 cm) in the block code. If the student wants the robot to move exactly 1, meter she will have to include 5 forward actions in her code (or a block that repeats 5 times). Likewise, the turn arrow represents turning a predefined angle (set to 90 degrees). The language also includes simplified blocks for conditionals as can be observed in Fig. 3a. This design decision means that a different conditional block needs to be created for each condition that wants to be tested. In the figure the condition is: is there something blocking the view of the robot? This leads to a proliferation of blocks but we thought it was a reasonable compromise for conditions that are frequent when teaching preliterate children.

`UNC++Duino` interface offers all programming languages in different tabs. The automated translation into Python or C++ occurs when the student changes to the Python or C++ tab and the interface detects that the block code has been modified. If the Python code is modified then it is also automatically translated into C++. The translation is not done backwards, it is not bidirectional. That is, if a student changes directly the code in Python or C++, the block code is not updated and a warning sign is showed in the interface. It is not possible to make the translation bidirectional because some of the programming languages used are less expressive than others. For example, in the iconic language it is not possible to express that the robot has to advance exactly 10cms. We informed the students in advance that the translation is not bidirectional, and they had no problem with it. We assume that children will eventually explore text languages, seeking to have more control of the robot functions and grow out of the iconic interface.

`UNC++Duino` is a web platform programed in JavaScript and runs in most web browsers. The interface is minimalistic, it shows only one programming language at a time, and the student can easily change from one language to another by tabs. It includes a button that sends the code to the robot and another one to share the code (e.g. with the teacher or a fellow student). The block languages include a menu of the available blocks that the student can drag and drop building the code as a puzzle.

Since `UNC++Duino` is open source, it allows users to create a new block, and to define its translation to Python and C++. We have defined already a set of blocks that are useful for the specific hardware we have been working. But, the software was designed to be easily extendable by adding new blocks as well as new programming languages other than Python or C++.

We can create a new block by performing the following two steps. First, we define the block; specifying its name, color, structure (that is, whether it requires some nested code such as the loop or the conditional) and its parameters. The new block definition can be directly written in JavaScript, referring by code to the jpg or png image(s) that represent the icon. When we created the if-then-else block shown in Fig. 3a we used two photos of the robot, one with and one without an obstacle. Instead of programming the new block, it can be designed graphically using the Blockly tool designed for this purpose.[4]

---

[4] https://blockly-demo.appspot.com/static/demos/blockfactory/index.html.

Second, we define the translation of the new block to the specific programming language we want. `UNC++Duino` already includes two files, the python.js and arduino.js, that are responsible for merging the code generated by the translation of a set of blocks into Python and C++. If we want to translate the new block (or an existing block) in another language, we also have to develop the merging code file. The following Python code is generated by python.js when automatically translating the code in Fig. 3a.

```
def avoidObstacles(robot):
while True:
     if robot.ping() <= 20:
         robot.turnLeft(60,0.5)
     else:
         robot.forward(60,0.5)
```

The robot's methods `ping`, `turnLeft` and `forward` chose the appropriate sensors or actuators to use. The method `ping` uses the ultrasound sensor to check that the road is clear for at least 20 cm; `turnLeft` rotates one wheel in order to turn left at 60rpm during half a second, and `forward` rotates both engines at 60rpm during half a second.

Once we have defined the block and generated the code, we have to the compile the new code. The interface will then be updated to include the new block or programming language.

5 Activities for Different Age-Groups in the Classroom

In this section we describe different activities developed in classroom with the `UNC++Duino` platform. Our iconic programming language, allows people from a wide range of ages to program: from preliterate children to students who are taking their first steps in programming. This high level language let students focus on the program structure rather than on the syntax and other technical details until they are ready to do so. Indeed, although originally designed for little children and evaluated with them [14], `UNC++Duino` proved to be engaging but also challenging for in-service K-12 teachers when taking their first steps in programming [13]. One of our goals of `UNC++Duino`, is that students can grow with the platform, so we describe here a set of activities that we designed in order to illustrate how to take advantage of the flexibility of the tool.

Consider again the challenge of programming a robot that could avoid objects. In the previous section we showed how this challenge can be solved using the iconic language and also the Blockly language (Fig. 3). We also showed how the same problem can be solved using Python. In the Python method `avoidObstacles` the students can directly modify the parameters to change, for instance, the speed of the robot, or the distance from the obstacles that the robot tolerates. They can also make the

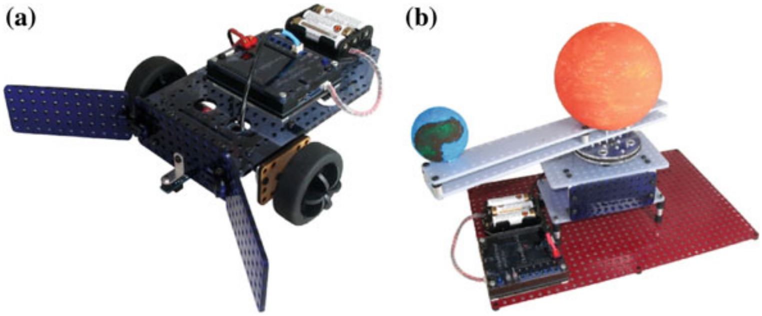


Fig. 4 A combine harvester that collects grain and a planetary system

robot move backwards before turning by sending a negative parameter to the method `forward` (something that will prove necessary if they define the ping distance too low).

Once students have programmed the robot that avoids objects, we could add an extra challenge: create a firefighter robot. Teachers and students should design a maze with different objects that the robot should avoid. In a part of the maze, the teacher turns on a candle. The robot has to find the candle and turn it off. The room has to be dark, because the robot can detect where the candle is using photosensors.

Instead of programming a robot that avoids objects, students could program one that collects them. With the robotic kit and a screwdriver students can construct a toy "harvester" similar to that illustrated in Fig. 4a that can "harvest" a predefined field inside the classroom collecting crumbled yellow paper balls representing the grain. Fields can be delimited with color tape so that the robot bottom sensors are used to stay inside. A competition between different students groups can be organized; the student who collects more grain wins.

Teachers from other disciplines could integrate Robotics and CS into their curriculum. A planetary project could be part of the activities. Using the educational kit, as illustrated in Fig. 4b, we can show our students a solar system structure, and tell the children that their challenge is to make it move like the sun and the earth move. We can introduce programming concepts such as *parameter* and *cycle*. Younger children would use iconic blocks, that would help them move the engines faster or slower, in order to experiment with different speeds.

High school students could program the same experience, but they could build the planetary system as well. In this way, they will interact directly with arduino board, sensors and engines. The programming goal would be to decompose the problem, and work with programming concepts such as *conditionals*, *cycles*, *methods*. They could also program the constants and calculations necessary to mimic the movements of the earth and the sun on a faster scale. Another challenge could be to turn on a led when the country they live in is not facing the sun.

A good idea could be to create something related with students life. Elevators make children feel curious about how they work. If we are in a mall or building,

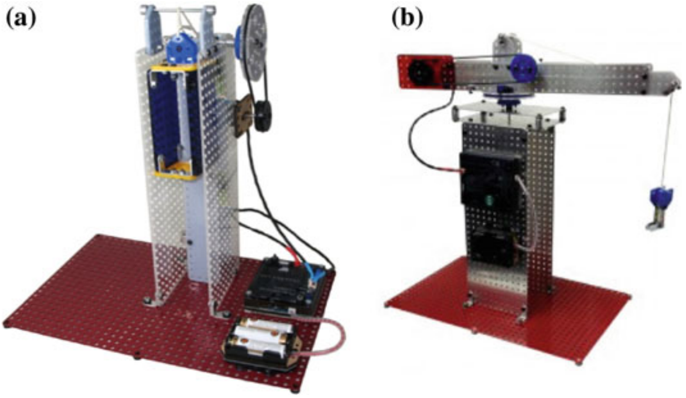


Fig. 5 An elevator programmed with a queue and a crane. **a** An elevator. **b** A crane

children want to use the elevator. Here we define a challenge to develop an elevator at school (Fig. 5).

For kindergarteners, we can talk about elevators and how, in general, they work. With the educational kit, teachers could build the elevator, and little children could program it with `UNC++Duino` iconic blocks. The elevator will be equipped with 3 buttons that students will activate when testing the elevator (ground floor, 1st floor and 2nd floor).

High schools students, should construct the elevator using the educational kit. Then, they will incorporate programming, using `UNC++Duino` language blocks to begin with. Students could use events for programming the elevator. To resolve the elevator problem, students will have to manage when different users of elevator touch different floors buttons. How the elevator should behave to be fair to the people waiting for it? More advanced students could move to Python and use the queue data structure.

6 Discussion and Conclusions

`UNC++Duino` has been evaluated by teaching programming to kindergarten and primary school children [14] as well as for in-service teacher training in a Professional Development program [13]. For precise evaluation data we refer the reader to these articles. Below we summarize their findings.

These studies found that most evaluated subjects, starting from 6 years-old, could learn fundamental programming concepts such as sequence, loops, parameters, conditionals through `UNC++Duino` and were capable of applying these notions into robot programming. However, older children (from 10 years-old) could combine these concepts better to create a new program [14]. The study described in [13] found that, after a 50-hour teacher training program, teachers could use the programming concepts but only those with previous background in CS were able to fully explain them to their students. The authors in [14] report that students showed high engagement and that `UNC++Duino` triggered exploration. Most students explored and commented on the different tabs with other programming languages and were able to notice the parallelism between the equivalent program structures. Indeed, some students were able to modify parameters in the text languages that were not modifiable in the block languages although the activity was not designed with this goal in mind.

In this paper we presented `UNC++Duino`, an open source educational software for learning to program a robotic kit in C++ and Python starting from drag-and-drop programming languages. One of the block based languages included is completely iconic allowing for its use with preliterate children as well as with beginners. Besides simplifying the initial steps, the code resulting from translating block code into text languages is designed to be highly modular, as illustrated in Sect. 4.

In general, children grow out of computer platforms designed for a particular age group very quickly, this is good for children but not so for teachers who sometimes cannot keep up learning to use different complex (and many times proprietary) interfaces. Our platform encourages students to grow with it and because of it. We acknowledge that further research is necessary with multi-language platforms for CS Education, but this, combined with robotics could be one direction to encourage children (and teachers) growth in CS.

Acknowledgments This work was partially funded by the grants PICT-2014-1833, PICT-2012-712, PDTS-CIN-CONICET-2015-172, and PID-2012-2013-R18.

References

1. Bell, T.: Establishing a nationwide CS curriculum in New Zealand high schools. Commun. Assoc. Comput. Mach. **57**(2), 28–30 (2014)
2. Benotti, L., Martínez, M.C., Schapachnik, F.: Engaging high school students using chatbots. In: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, ITiCSE'14, pp. 63–68. ACM, NY, USA (2014)
3. Bergen, D.: Technology in the classroom: learning in the robotic world: active or reactive? Child. Educ. **77**(4), 249–250 (2001)
4. Carter, L.: Why students with an apparent aptitude for computer science don't choose to major in computer science. SIGCSE Bull. **38**(1), 27–31 (2006)
5. Clements, D.H., Sarama, J.: Teaching with computers in early childhood education: strategies and professional development. J. Early Child. Teach. Educ. **23**(3), 215–226 (2002)
6. Dagiene, V., Mannila, L., Poranen, T., Rolandsson, L., Söderhjelm, P.: Students' performance on programming-related tasks in an informatics contest in Finland, Sweden and Lithuania. In: Proceedings of the 2014 Conference on Innovation; Technology in Computer Science Education, ITiCSE'14, pp. 153–158. ACM, NY, USA (2014)
7. Flannery, L.P., Bers, M.U.: Let's dance the "robot hokey-pokey!" Children's programming approaches and achievement throughout early cognitive development. J. Res. Technol. Educ. **46**(1), 81–101 (2013)

8. Flannery, L.P., Silverman, B., Kazakoff, E.R., Bers, M.U., Bontá, P., Resnick, M.: Designing ScratchJr: support for early childhood learning through computer programming. In: Proceedings of the 12th International Conference on Interaction Design and Children, pp. 1–10. ACM (2013)
9. Furber, S.: Shut down or restart? The way forward for computing in UK schools. Technical report, The Royal Society, London (2012)
10. Grover, S., Pea, R., Cooper, S.: Remedying misperceptions of computer science among middle school students. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE'14, pp. 343–348, ACM, New York, NY, USA (2014)
11. Kazakoff, E., Sullivan, A., Bers, M.: The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Child. Educ. J. **41**(4), 245–255 (2013)
12. Magnenat, S., Shin, J., Riedo, F. Siegwart, R., Ben-Ari, M.: Teaching a core CS concept through robotics. In: Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education, ITiCSE'14, pp. 315–320. ACM, New York, NY, USA (2014)
13. Martinez, C., Gomez, M., Benotti, L.: Lessons learned on computer science teachers professional development. In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE'16. ACM, NY, USA. In press
14. Martinez, C., Gomez, M.J., Benotti, L.: A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE'15, pp. 159–164. ACM, New York, NY, USA (2015)
15. Meerbaum-Salant, O., Armoni, M., Ben-Ari, M.M.: Learning computer science concepts with Scratch. In: Proceedings of the Sixth International Workshop on Computing Education Research, ICER'10, pp. 69–76. ACM, NY, USA (2010)
16. Morgado, L., Cruz, M., Kahn, K.: Preschool cookbook of computer programming topics. Australas. J. Educ. Technol. **26**(3), 309–326 (2010)
17. Morgado, L., Kahn, K.: Towards a specification of the ToonTalk language. J. Vis. Lang. Comput. **19**(5), 574–597 (2008)
18. Perdikuri, K.: Students' experiences from the use of MIT app inventor in classroom. In: Proceedings of the 18th Panhellenic Conference on Informatics, PCI'14, pp. 41:1–41:6. ACM, New York, NY, USA (2014)
19. Portelance, D., Strawhacker, A., Bers, M.U.: Constructing the scratchjr programming language in the early childhood classroom. In: International Journal of Technology and Design Education, pp. 1–16 (2015)
20. Wilson, C.: Running the Empty: Failure to Teach K-12 Computer Science in the Digital Age. Association for Computing Machinery (2010)
21. Wilson, C.: Hour of code: we can solve the diversity problem in computer science. ACM Inroads **5**(4), 22–22 (2014)
22. Zur Bargury, I.: A new curriculum for junior-high in computer science. In: Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education, pp. 204–208 (2012)

Usability Evaluation of a Raspberry-Pi Telepresence Robot Controlled by Android Smartphones

Krit Janard and Worawan Marurngsith

Abstract Telepresence robots in static pan-tilt form can be a viable and affordable choice for tele-education. However, cost considerations may impose limitations on usability and expected performance. The goal of this study was to explore the usability of a low-cost, static pan-tilt telepresence robot operated using an Android smartphone. Experiments were carried out with 26 participants from two age groups (14 students M = 20 years, 12 staff M = 25 years) in a laboratory. Each participant interacted with the robot to perform two tasks. The opinions of the participants pre- and post-experiments, and the time they took to complete the tasks, were recorded. The results show that the average latency (of 3.1 ± 0.8 s for one robot movement) is quite acceptable. The students were faster than the staff when controlling the robot remotely but slower when working at the robot site. Correlation analysis shows that confidence in the robot and the likelihood of adoption is strongly related to data privacy features. All the methods used to control the robot remotely show positive interaction to each other. This implies that the majority of participants were focussed on the control methods and data privacy provided in the robot platform, and were willing to accept a small delay in robot movement.

Keywords Usability testing • Performance evaluation • Telepresence robot • Raspberry pi • Higher education

1 Introduction

Telepresence robots already play a large role in telemedicine, tele-education and remote offices for enterprises, as they allow users to have ad hoc verbal communication in a remote, restricted, underserved or dangerous area. Telepresence robots have been successfully used to extend accessibility to remote education around the

---

K. Janard · W. Marurngsith (✉)
Faculty of Science and Technology, Department of Computer Science, Thammasat University, Bangkok 12120, Pathum Thani, Thailand
e-mail: wdc@cs.tu.ac.th

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_18

world [1–6]. Some common uses are allowing students with special needs to attend classes and to perform laboratory experiments remotely, allowing remote teachers to give lectures to underserved or restricted areas, and conducting field trips (as reviewed in [7, 8]). Despite their broad range of applications, commercial robots are not affordable for many institutions. Thus, several affordable robots based on Raspberry Pi (Raspi) computers, with essential autonomy over e.g., head movement using a pan-tilt unit, have been proposed [9, 10]. Applications involving head movements have shown positive impacts on user involvement when interacting with telepresence robots [11]. Thus, a pan-tilt telepresence robot with the capability to orient its screen to face the interlocutor can be a good candidate for educational use.

Although telepresence robots in static pan-tilt form can be an affordable choice for remote learning, cost considerations may impose issues on usability. As reviewed in [12], usability concerns not only the effort needed to use a system, but also the extent to which the system may be used by specific users in a specific situation to achieve particular goals. Usability covers the user's experience before, during and after the interaction with the system. The focus of a robot's usability is on the control method and the interaction between the human and the robot. Several control methods have been developed, along with evaluations of their usability e.g., using wheelchair control [13], using tricycle-style control [3], control by tablets [14], and control by smartphones [15, 16]. Among these investigations, the usability of robots by school children has been addressed in [3]. To use a static pan-tilt telepresence robot effectively in a higher educational context, a systematic usability evaluation is needed.

The goal of the study described in this paper was to explore the usability of an affordable, static pan-tilt telepresence robot (Fig. 1), called ACTR (the Android Controlled Telepresence Robot) in Higher Education. The robot used a low-cost

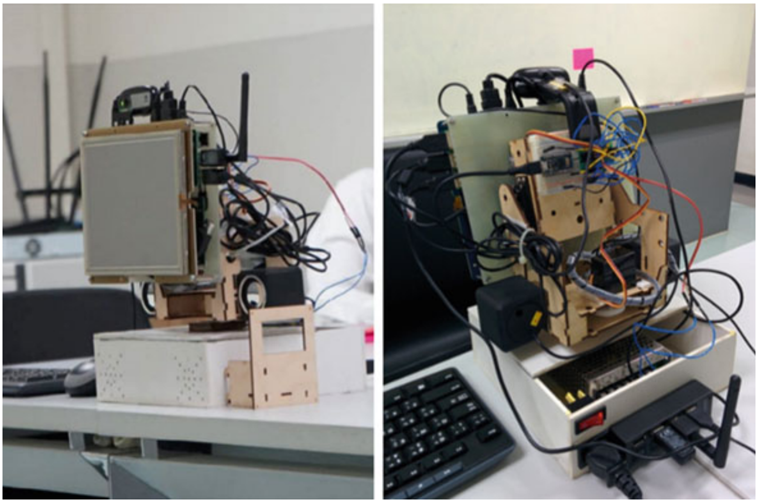


Fig. 1 Prototype of the ACTR robot

Raspi2B computer as its main controller. Thus, the cost to develop the robot prototype was US$600. To explore the usability of the ACTR, usability testing and assessment were performed in a laboratory. Experiments were carried out on twenty-six volunteers from two age groups (14 students M = 20 years., 12 staff M = 25 years). Each participant performed two tasks. In the first task, the participants stayed at the remote site. He/she used an appropriately programmed smartphone to remotely control the robot using three control methods (i.e., using navigation buttons, tilting the smartphone, or using an automatic face-following mode). The aim of this task was to observe how the participants controlled the ACTR robot to accomplish the task. In the second task, the participants stayed at the robot site but had a discussion with an interlocutor via the robot. During this task, the participants were asked to walk around so that the robot could turn its head to follow the face of the participants. Empirical data, including the satisfaction rating from the participants pre- and post-experiments and the time to complete each experiment, were kept and analysed.

The results showed that a majority of participants agreed that the speed of the low-cost robot and the ways of controlling it, were acceptable. The majority of the participants also agreed that the robot could be integrated well into educational use. Correlation analysis showed that users' confidence in interfacing with the robot, and their intention to adopt one, is strongly correlated with the data privacy feature. Moreover, the robot's design should focus on an efficient and effective control method in order to ensure a satisfactory user experience.

2 Android-Controlled Telepresence Robot Using Raspberry Pi

2.1 The ACTR Prototype

An Android-controlled telepresence robot based on a Raspi2B computer, called ACTR was used in this study. The ACTR robot, the upgraded version of [10], is a static pan-tilt robot which can rotate its display to follow the face of a speaker who moves during a talk. This face-following capability imitates the natural synchronous posture of a listener when he/she pays attention to the speaker. The robot's hardware consists of a monitor, two cameras (i.e., a webcam and a Raspi camera module), two servo motors forming a pan-tilt unit (PTU), and an AC power supply (see Fig. 1). The Raspi2B computer and an Arduino Nano controller were used for real-time face detection and video conferencing simultaneously. The Local Binary Pattern (LBP) algorithm from the Open Source Computer Vision (OpenCV) library was used for face detection. The computation was accelerated by offloading to the GPU of the Raspi using the multimedia abstraction layer (MMAL) application program interface.

2.2 ACTR Robot Usage Scenario

In Higher Education, the ACTR robot can be used for remote learning, remote teaching and group discussion. The most common use is for remote learning in which the robot can be used to allow a student to attend a class remotely via video conferencing. In this case, a lecturer teaches the class at the robot site and interacts with the robot. The student remotely connects to the robot and takes control of the robot by using an application provided for any Android smartphone. The connection between the robot and the smartphone is done by using a virtual private network (VPN) over the Internet to ensure data privacy. Once the connection has been established, two modes of communication operate concurrently. First, manual control of the robot PTU runs in background mode to control the head position. Second, the webcam interface runs in foreground mode to display the images of the student and the lecturer on the screen of the robot and the smartphone. The student can set the robot to freeze at its current position or control the movement of the robot by using three control methods. These are (1) using the navigation buttons, (2) tilting the smartphone or (3) using the autonomous face-following mode.

2.3 User Interface Design

Since the ACTR robot and the smartphone cooperate over the Internet, the design of the user interface (UI) also considered issues regarding the user experience of a system with multiple devices. The key issues considered in the UI design are the conceptual model, usability, consistency, continuity, latency and reliability, as suggested in [17]. In terms of the conceptual model, the UI of the robot and the smartphone have an icon to show the status of the remote device (Fig. 2A and B).

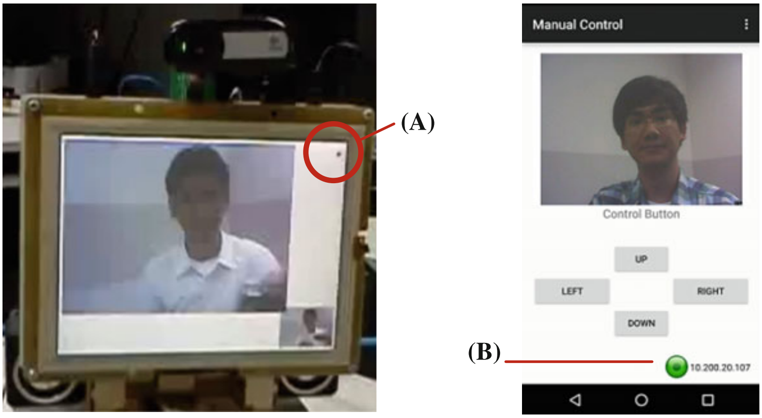


Fig. 2 User interface on the robot's display (*left*) and on the smartphone (*right*)

This is to allow users to clearly see the overall connection of the system conceptually, and to see what might have gone wrong if the connection is not successful. For interoperability, all of the control functionality is implemented on the Android device as it is a suitable device in the context of use. Consistency and continuity require the use of consistent naming for the same system features, and the ability to continue the task on other devices respectively. Since the ACTR framework has only one control device, the UI design does not have to cover these issues. Regarding latency and reliability, a transparent model was used. This means that when a remote user tries to connect to the ACTR robot, the UI will show the real situation while it is trying to connect. The connection light is red when the robot is not connected and blinks while it is trying to establish a connection. The connection light turns green, when the robot has sent back an acknowledgement message.

3 Design of Usability Evaluation Experiments

In order to improve user interaction, empirical evidence on how users use the ACTR robot to execute tasks was gathered by performing usability evaluations. The relevant ISO standard distinguishes five usability factors: efficiency, effectiveness, satisfaction, absence of risk, and context coverage.[1] Efficiency means users can perform the key tasks within an acceptable time interval, while effectiveness also specifies the level of satisfaction of the users to finish the tasks. To examine usability of the ACTR robot in a Higher Education context, both usability testing and assessment methodologies [12] have been used. The goal was to systematically observe how well students and staff can use the robot for discussion in a lab.

3.1 Participants

Twenty-six volunteers participated in the experiments i.e., 14 students (7 females, 7 males M = 20 years.) and 16 staff (5 females, 11 males M = 25 years.). Their subjective experience and characteristics are summarised in Table 1. The average height of participants is 167.7 cm. (approximately 30 cm different between the tallest and the shortest). Most of the participants made video calls three times a month on average, with each call lasting for approximately twenty eight min. Very few participants engaged in teleconferencing. Many participants used Android phones, and thought that smartphones were the main device for making video calls. However, most of them only used computers to do an online video conversation.

---

[1] The original ISO 9241-11 standard defines efficiency, effectiveness and satisfaction. Later in 2011, the ISO 25010 added absence of risk, and context coverage to the ISO 924-11.

Table 1 Measures of subjective experience and characteristics of participants

	Factor	Range	$\bar{x}$	SD
1	Height (cm)	154–183	167.7	7.9
2	Frequency of video call usage (times/month)	0–6	3	2.2
3	Average length of video call usage (minutes/call)	0–60	28.3	14.6
4	Frequency of teleconference usage (times/month)	0–6	0.79	1.43
5	Average length of teleconference usage (minutes/call)	0–60	20.77	22.82
6	Mobile phone's OS (Android:iOS:Windows)	17:07:02		
7	Awareness of devices used for video calls (Computer:Smartphone:Tablet:Other)	17:23:11:1		
8	Experienced of video calls on devices (Computer:Smartphone:Tablet:Others)	12:2:0:1		

3.2 Method

The local usability testing and assessment were done in a laboratory comprising two rooms, one emulating the robot site and the other the remote site. For the usability testing, all participants performed two sets of predefined tasks with the robot developer. In the first task, the participant stayed at the remote site while the robot developer stayed at the robot site. The participant used a prepared smartphone to control the robot remotely while pretending to discuss laboratory work with the robot developer. The participants were asked to control the ACTR robot using three control methods i.e., (1) using navigation buttons, (2) tilting the smartphone, or (3) using the automatic face-following mode, using one method at a time. The aim of this task was to measure the time and perception of the participants in turning the head of the ACTR robot to face several marked points at the robot site. The participants were asked to complete this task in three different ways i.e., (1) moving the robot's head in the horizontal and vertical (H and V) directions, (2) moving the robot's head in a diagonal direction, (3) allowing the robot's head to follow the face of the interlocutor autonomously.

In the second task, the participants stayed at the robot site, pretending that the participants had to give a talk to the remote user. In this task, the participants were asked to walk around, while the listener at the remote site configured the robot to perform autonomous face following.

For the usability assessment, the opinions of the participants pre- and post-experiments, and the time to complete each scenario were recorded.

3.3 Measurement and Data Analysis

The amount of time the participants required to accomplish each task was recorded in units of seconds. The participants' behaviour was video recorded, with the participants' consent. Data from questionnaires were also collected pre- and post-experiments. The questionnaire had five parts. The first part had sixteen questions on demography and prior experience. Later parts had satisfaction rating (using the 1–5 Likert scale). The last section of the questionnaire had a box for written comments.

All the empirical data gathered from the experiments were entered into an Excel spreadsheet and analysed in five steps. First, the data obtained from the students and staff were tested to see whether they showed different responses, using the two-sample T-test. Second, the one factor ANOVA test was used to determine whether there were any significant differences between the means of data when steering the robot in different directions using one control method. Third, the average time and satisfaction rate on the Likert scale were summarised by using an arithmetic mean with 95 % confidence interval and presented in forest plots. Fourth, the correlation of features related to the participants' acceptability of the robot were analysed by constructing a correlation matrix. Lastly, the change of satisfaction rating scores before and after the experiments were analysed by calculating the difference and using statistical summary.

4 Experimental Results and Discussion

4.1 Efficiency

The efficiency of the ACTR robot was measured by observing the average *time to complete tasks*. All participants were able to finish all tasks successfully but the results from the students and the staff were significantly different. As shown in Fig. 3, the students used less time to finish the tasks when controlling the robot remotely. The students were 20 % and 10 % faster than the staff using the button and phone-tilting control methods respectively. Phone-tilting control allowed the tasks to be finished faster than button control at 44 % and 58 % for the students and staff respectively. However, as shown in Fig. 4, the students were slower and showed more variation in the total time to finish the task when they were communicating with the robot site, approximately 10 % slower. The total time shown in Figs. 3 and 4 is the time taken to move the robot twenty times. On average, the latency was 3.1 ± 0.8 s. The results clearly show the greater efficiency of the phone-tilting control method over button control.

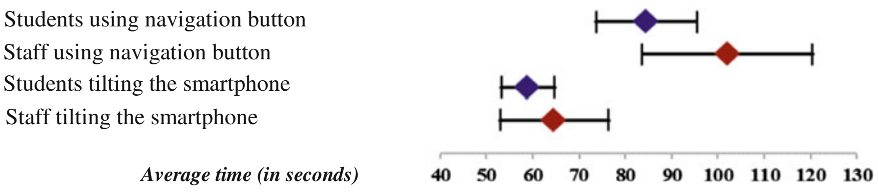


Fig. 3 Average time to complete the tasks of participants using different control methods

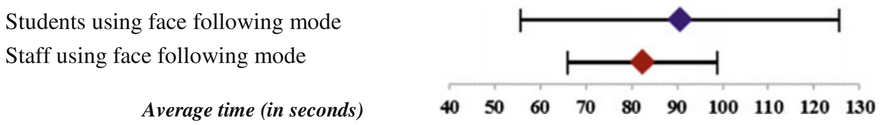


Fig. 4 Average time to complete the tasks of participants using automatic face-following

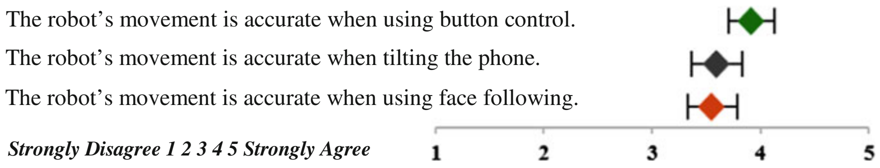


Fig. 5 Perception of accuracy when controlling the robot in horizontal and vertical movement

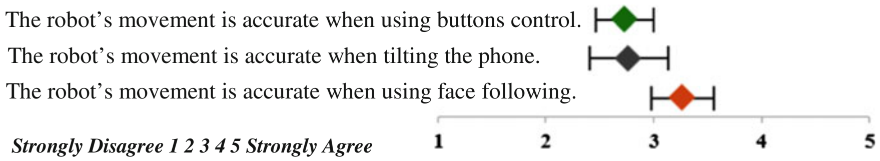


Fig. 6 Perception of accuracy when controlling the robot in diagonal movement

4.2 Satisfaction

The opinions of students and staff showed no statistical difference when moving the robot in different directions. However, the results obtained from moving horizontally and vertically (Fig. 5) were different from moving diagonally (Fig. 6). Interestingly, participants gave the button control the highest score for accurately moving the robot in the H and V direction (avg. of 3.9). However, button control was perceived as the least accurate method when moving the robot diagonally (avg. of 2.7). Button control also received the lowest score for the robot's responsiveness, i.e., having an average rating of 2.9 (see Fig. 7). The smartphone-tilting control method also received a similar rating pattern, yet having a narrower range of rating scores (i.e., average rating of 3.6 and 2.8 between moving in H and V and diagonal

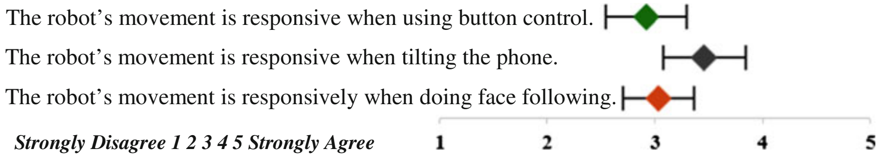


Fig. 7 All participants' perception score on responsiveness of the robot using different modes

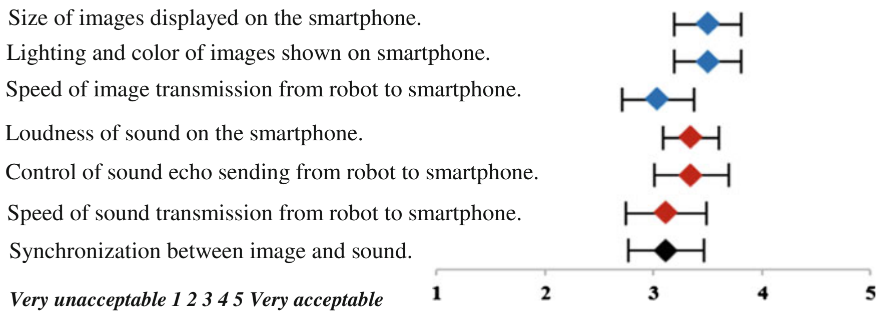


Fig. 8 Acceptability rating of image and sound quality at the smartphone site

directions respectively). However, tilting the smartphone was rated as the most responsive method (see Fig. 7, avg. of 3.5). The participants felt that accuracy dropped when steering the robot to move diagonally (see Fig. 6 in comparison to Fig. 5). When ranking the average participants' rating of different control methods from the highest to the lowest score, the ranking is phone tilting, face following and buttons.

4.3 *Acceptability*

Acceptability was analysed using three criteria (a) the Likert-scale acceptable rating of video conferencing at the remote site, (b) the correlation of features affecting the participants' acceptability rating and (c) the change of satisfaction rating after using the robot. As shown in Fig. 8, the participants found that the image and sound quality fell between neutral to slightly acceptable. Correlation analysis shows that the data privacy feature is strongly correlated with confidence, and the likelihood of adopting the robot. As shown in Table 2 in bold face, the look of the hardware design was also very much related to the likelihood of adopting the robot. After using the robot, most participants changed their opinions on five aspects (see

Table 2 Correlation matrix showing the related features in bold face

	Confidence	Data privacy	HW look	UI look	Adoption likelihood	Should cont.
Confidence	1					
Data privacy	**0.706**	1				
HW look	0.094	0.081	1			
UI look	0.192	0.218	0.287	1		
Adoption likelihood	**0.439**	0.312	**0.391**	0.315	1	
Should cont.	0.024	0.157	0.140	0.163	**0.721**	1

Table 3 Statistical summary showing the change of satisfaction rating after using the robot

Students (Age 18–22)					Staff (Age 23–40)				
Strongly Disagree 1 2 3 4 5 Strongly Agree (satisfaction rating using the 1–5 Likert scale)									
Before		After	Changes		Before		After	Changes	
(1) *Participant feels confident in using smartphone to control the robot*									
Range	2–5	2–5	%	57 %	Range	2–5	3–4	%	**92 %**
Average	3.3	3.3	+	38 %	Average	3.2	3.7	+	73 %
Std.dev.	0.83	0.91	–	63 %	Std.dev.	0.83	0.49	–	27 %
(2) *Using a telepresence robot is becoming common*									
Range	1–5	1–5	%	43 %	Range	2–5	3–4	%	33 %
Average	3.2	3.6	+	**83 %**	Average	3.4	3.7	+	75 %
Std.dev.	1.12	1.22	–	17 %	Std.dev.	0.79	0.49	–	25 %
(3) *Using the robot can increase the quality of interaction*									
Range	2–5	2–5	%	64 %	Range	2–4	3–5	%	33 %
Average	3.3	3.9	+	78 %	Average	3.6	4.3	+	**100 %**
Std.dev.	1.07	1.03	–	22 %	Std.dev.	0.67	0.62	–	0 %
(4) *The price of the robot (US$600) is affordable for educational use*									
Range	1–4	2–5	%	29 %	Range	2–4	3–5	%	67 %
Average	3	3.5	+	**100 %**	Average	3.2	3.8	+	**88 %**
Std.dev.	0.96	0.85	–	0 %	Std.dev.	0.72	0.58	–	13 %
(5) *The functionality and utilisation of the robot meets your expectation*									
Range	2–5	2–5	%	57 %	Range	3–5	3–5	%	42 %
Average	3.8	4.1	+	63 %	Average	4.1	4.2	+	60 %
Std.dev.	0.97	1	–	38 %	Std.dev.	0.67	0.58	–	40 %

Table 3). The staff's opinions were less varied than the students. After using the robots, all participants showed more positive rating for the affordability of the robot in educational use and agreed that using the robot can increase the quality of interaction. A majority of the participants showed more positive agreement that using telepresence robots is becoming common. After doing the experiments, some of the students seemed to have less confidence to control the robot.

4.4 Discussion

The results confirmed that using a static pan-tilt telepresence robot in Higher Education is efficient and effective. The participants were quite satisfied with having three ways to control the robot. When informed of robot price, the participants felt that the quality of the image and sound are quite acceptable. This shows that within this price range, the participants did not expect much improvement in the video conferencing quality. After using the robot, many participants felt less confidence to control the robot. This suggested that the expectation of participants was different from what the system provided. Thus, the effectiveness and ease of use of the control methods should be improved. As the experiments were done in a local setting, the real communication delay has not yet been observed. Thus, more work should be carried on to assess the impact of internet delays on the quality of the communication offered via the robot.

5 Related Work

The PEBBLES mobile telepresence robot is one of the earliest developments of a robot for students at elementary school and high school level used in Canada and the USA [1]. The high school version of the robot has two monitors to display the student's face and his/her work separately. The unique feature of the robot is a wire hand for waving and getting attention. The robot prototype has been tested with users, however no usability study has been reported. A commercial robot, VGo, has also been used to enhance the educational experience of a special needs student [2]. The case study showed that the student felt that the robot has given him back his socialization and excitement about school. Moreover, the cost of using the robot (at US$6,000) was only one tenth of the cost to provide private lessons for students with special needs.

In Japan, a child-operated telepresence robot using tricycle style has been proposed for early education [3, 6]. Usability testing on twenty children of 4–8 years old was used to compare two controlling styles i.e., using a tricycle device and using a video game controller. The results showed that the children performed tasks faster when using the tricycle interface. This research emphasizes that the interface used to control a remote robot must suit the physical and mental state of the expected users.

Instead of a student-controlled telepresence robot, in Korea, the teacher-controlled, mobile Tele-education robot has been proposed to offer an English language course [4]. A 12-week pilot project was performed in 2011 using 29 English teacher robots controlled by native teachers in the Philippines, with each teacher conducting a class of 8 students. The results show that majority of students found the class useful, had a strong desire to re-attend the class and were more interested in learning English.

An affordable telepresence robot for education at university level, with an arm capable of performing remotely in a chemistry laboratory, has been proposed in [5]. The robot was built within the budget of $400 Canadian. The case study showed that remote laboratory and telepresence robots can provide added flexibility to online learning and can allow special needs students to experience learning activity in real-time.

The usability of telepresence robots has been addressed in many publications. Yan et al. studied the usability of a pan-tilt static robot with the capability to orient its display in the direction of a user voice [18]. Experiments performed on 15 participants showed that the top rank usage scenario suggested by the participants is for a business meeting. Lewis et al. evaluated the usability of two commercial telepresence robots, the VGo (US$6,000) and AVA500 (US$70,000) [19].

6 Conclusion and Future Work

Telepresence robots have been applied to education at every level ranging from a well-equipped, full function commercial robot for business to an affordable design for Higher Education. Telepresence robots in static pan-tilt form can be an affordable choice to integrate with existing online activities for Higher Education. Usability evaluation is a key step to gather empirical evidence about how users use the robot. Therefore, within the cost limitation, the robot's design can be suitable and improve user interaction. This study aimed to examine the usability of a low-cost, static pan-tilt telepresence robot controlled using Android smartphones. Usability testing and assessment were carried out on 26 students and staff. Empirical data, including, the time to complete tasks and subjective satisfaction ratings were gathered and analysed. The results showed that the participants had neutral to positive attitude to the performance, and to the ways to control the robot. A majority of the participants agreed that the robot could integrated well into educational use. The results of correlation analysis highlighted that the confidence to interface with the robot, and the intention to adopt one, is strongly related with data privacy feature. Thus, future work will focus on enhancing the data privacy feature and improving the degrees of autonomy such as mobility, turning the robot's head by voice or by poking, calling for attention by raising a hand, or a light signal.

Acknowledgments This project has been funded by the Faculty of Science and Technology, Thammasat University. We thank the reviewers for their valuable comments. We thank contributors to the Raspberry Pi community website for discussions and lesson learned. We thank Professor Roland Ibbett for improving the readability of this paper.

References

1. Yeung, J., Fels, D.I.: A remote telepresence system for high school classrooms. In: Canadian Conference on Electrical and Computer Engineering (2005)
2. Bloss, R.: High school student goes to class robotically. Ind. Robot Int. J. **38**, 465–468 (2011)
3. Tanaka, F., Takahashi, T., Morita, M.: Tricycle-style operation interface for children to control a telepresence robot. Adv. Robot. **27**, 1375–1384 (2013)
4. Yun, S.-S., Kim, M., Choi, M.-T.: Easy interface and control of tele-education robots. Int. J. Soc. Robot. **5**, 335–343 (2013)
5. Denojean-Mairet, M., Tan, Q., Pivot, F., Ally, M.: A ubiquitous computing platform—affordable telepresence robot design and applications. In: 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE), pp. 793–798 (2014)

6. Tanaka, F.: Robotics for Supporting Childhood Education. In: Cybernics: Fusion Of Human, Machine And Information Systems, pp. 185–195. Springer, Tokyo (2014)
7. Tsui, K.M., Yanco, H.A.: Design Challenges and Guidelines for Social Interaction Using Mobile Telepresence Robots. In: Reviews of Human Factors and Ergonomics, vol. 9, pp. 227–301 (2013)
8. Kristoffersson, A., Coradeschi, S., Loutfi, A.: "A review of mobile robotic telepresence. Adv. Hum.-Comput. Interact. **2013**, 3 (2013)
9. Prabha, S.S., Antony, A.J.P., Meena, M.J., Pandian, S.R.: Smart cloud robot using raspberry Pi. In: 2014 International Conference on Recent Trends in Information Technology, ICRTIT 2014 (2014)
10. Janard, K., Marurngsith, W.: Accelerating real-time face detection on a raspberry pi telepresence robot. In: 2015 Fifth International Conference on INTECH (2015)
11. Bamoallem, B.S., Wodehouse, A.J., Mair, G.M., Vasantha, G.A.: The impact of head movements on user involvement in mediated interaction. Comput. Hum. Behav. **55**, 424–431 (2016)
12. Isabel, A., Queirós, M.A., Silva, A.G., Pacheco Rocha, N.: Usability Evaluation Methods: A Systematic Review. In: Saeed, S., Bajwa, I.S., Mahmood, Z. (eds.) Human Factors in Software Development and Design. IGI Global (2014)
13. Gironimo, G., Matrone, G., Tarallo, A., Trotta, M., Lanzotti, A.: A virtual reality approach for usability assessment: case study on a wheelchair-mounted robot manipulator. Eng. Comput. **29**, 359–373 (2012)
14. Clotet, E., Martínez, D., Moreno, J., Tresanchez, M., Palacín, J.: Development of a High Mobility Assistant Personal Robot for Home Operation. In: Ambient Intelligence—Software and Applications, vol. 376. Springer (2015)
15. Emami, M., Bahmani, M.R.: Design and implementation of a robot controlled by android. Int. J. Appl. Eng. Res. **10**
16. Tetzlaff, T., Zandian, R., Drüppel, L., Witkowski, U.: Smartphone controlled robot platform for robot soccer and edutainment. Advances in Intelligent Systems and Computing, vol. 345, pp. 505–518 (2015)
17. Rowland, C., Goodman, E., Charlier, M., Light, A., Lui, A.: Designing Connected Products: UX for the Consumer Internet of Things. O'Reilly Media (2015)
18. Yan, R., Tee, K.P., Chua, Y., Huang, Z.: A user study for an attention-directed robot for telepresence. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7910 LNCS, pp. 110–117 (2013)
19. Lewis, T., Drury, J.L., Beltz, B.: Evaluating mobile remote presence (MRP) robots. In: Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, pp. 302–305 (2014)

On the Design and Implementation of a Virtual Machine for Arduino

Gonzalo Zabala, Ricardo Moran, Matías Teragni and Sebastián Blanco

Abstract Arduino has become one of the most popular platforms for building electronic projects, especially among novices. In the last years countless tools, environments, and programming languages have been developed to support Arduino. One of these is Physical Etoys, a visual programming platform for robots developed by the authors. Physical Etoys supports compiling programs into the arduino. For this to work, a Smalltalk to C++ translator has been built. Although it has been very useful, this translator has brought a new set of issues. In this paper we will discuss some of these problems and how we decided to overcome them by developing a simple virtual machine that will be used as the base for the new Physical Etoys.

Keywords Arduino · Programming language · Virtual machine · Concurrency · Physical etoys

1 Introduction

Since the emergence of the Arduino board, the world has seen a significant increase in the amount of people without technical training (artists, designers, hobbyists) that have started to explore the world of microcontroller programming. The educational field has not been exempt of this trend. Following the movement that promotes

---

G. Zabala (✉) · R. Moran · M. Teragni · S. Blanco
Universidad Abierta Interamericana, Buenos Aires, Argentina
e-mail: gonzalo.zabala@uai.edu.ar

R. Moran
e-mail: ricardo.moran@uai.edu.ar

M. Teragni
e-mail: matias.teragni@uai.edu.ar

S. Blanco
e-mail: sebastian.blanco@uai.edu.ar

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_19

teaching programming and computer science in schools, robotics offers a highly motivational medium to introduce concepts from several disciplines. Furthermore, the amount of knowledge necessary to carry out small to medium size robotic projects is increasingly smaller. The Arduino board, being open hardware and low cost, invites students from all over the world to embark on the adventure of educational robotics.

The Arduino platform provides a simplified environment (based on the C++ programming language) in which most of the advanced microcontroller concepts are hidden away from the user. However, this environment is still too complex for some of the most inexperienced users, especially young children.

One of the main issues we found while teaching robotics to high school students is the limited support for concurrency in the Arduino programming language. Robotics competitions and exercises usually require the coordination of several concurrent tasks to achieve a goal. For instance, a common competition is the robot-sumo, in which two robots try to push each other out of a circle. This competition involves performing two simultaneous tasks: finding the opponent, and avoiding being pushed out of the circle. Other competitions involve more complex tasks. Most of the students participating in this kind of activities struggle to specify the concurrent behavior and as a result their robots perform poorly. The experience, thus, becomes frustrating instead of fun and engaging.

For these reasons, and taking advantage of the fact that Arduino is open source, there have been multiple attempts to provide a programming environment more suitable for beginners. One of these attempts is Physical Etoys (PE), an extension of Etoys [1] designed to provide children the tools to easily program different hardware platforms, including Arduino. Etoys is a media-rich authoring environment and visual programming system that allows children of all ages to create simulations and small videogames using a tile-based scripting system. Up to the creation of PE, Etoys only allowed to manipulate virtual objects, such as drawings on the screen. By using PE, a child can interact as easily with the virtual world provided by Etoys as with the real world through any of the supported robotic kits.

Etoys (and by extension, PE) provides a very simple concurrency model in which all of the user scripts run at configurable intervals inside an infinite loop. Although execution is entirely single-threaded, from the user perspective all the scripts are running simultaneously. This simple model is usually good enough for most students.

However, PE required the Arduino board to be constantly connected to the computer, which is not allowed in some robotics competitions. In order to address this issue PE can now also compile its scripts and upload them to the Arduino using a special programming mode. This feature involves a great deal of complexity that beginners are not usually able to handle. And since the PE firmware cannot coexist with the compiled programs, once the scripts are uploaded to the Arduino all the benefits provided by PE interactive environment are lost.

Moreover, this “compiled” mode has severe technical issues. In order for the compilation to work, PE first translates the scripts (which are automatically generated Smalltalk code) to C++, then it uses avr-gcc to compile the C++ sources into machine code, and finally it uploads the machine code to the Arduino board using avrdude. This process is highly complex and slow. It forces the PE distribution to include the AVR tools, increasing its size up to 5 times (from 47 to 231 Mb in its last version). Furthermore, the AVR tools are platform dependent, which makes it difficult to provide a single cross-platform distribution for PE.

Even though the ability to choose the programming mode depending on the kind of project being carried out has distinguished PE from other similar projects (such as Scratch for Arduino [2] or Minibloq [3]) the above-mentioned problems require a different approach.

The following requirements have to be met:

1. The script execution must be performed directly on the Arduino board without the need for interaction with the computer.
2. If the arduino happens to be connected to the computer, all the interactivity features provided by PE must be preserved.
3. The user should not be required to specify which programming mode to use (either compiled or interactive).

With these objectives in mind, it was decided to implement a virtual machine that could interpret the bytecode of a very simple programming language. This virtual machine (which was called Uzi) would be uploaded to the Arduino board once. The computer can then send individual instructions or entire programs for the Arduino to run by communicating with the virtual machine through the USB port.

In this paper we will discuss the design and implementation of the Uzi virtual machine, and we will compare it with other similar technologies in order to highlight the benefits and limitations of this solution.

2 Related Work

The development of virtual machines and high-level languages for small microcontrollers is not new. There have been a lot of attempts to provide a different programming environment for Arduino. Most of them are based on pre-existent general purpose programming languages such as Java, Scheme or Python.

HaikuVM is one of such attempts [4]. It is a Java VM based on leJOS [5] that runs on Arduino. Its compiler analyzes the Java source code in order to generate a C program that contains the user program (stored in Flash memory as a set of C structs) and the virtual machine that will interpret it. The user must then use the Arduino toolset to upload the program to the board. This implementation has benefits, such as the low memory usage by storing the user program in the Flash memory alongside the VM, but it needs the arduino tools to compile and upload the programs. The fact that it outputs C code allows the compiler to easily introduce special constructs that let the user inline C code, thus allowing him to choose the level of abstraction required for the problem at hand. HaikuVM supports almost all of Java semantics, including garbage collection, threads, and exceptions, but it lacks support for reflection, object finalization, weak references, and type information for arrays. In order to efficiently use the available memory in Arduino, the compiler performs a static program analysis that allows it to discard unused classes and thus generate more compact programs. Regarding performance, some benchmarks show an execution speed of “about 55k java opcodes per second on 8 MHz AVR ~ATmega8” [4].

Ocamm-pi is a variant of the Ocamm programming language [6] that supports several platforms, including Arduino [7]. Ocamm is especially designed to write concurrent programs, which are difficult to express using the Arduino language. It requires a board with at least 32 KB of space for code and 2 KB of RAM, so the smallest Arduino boards supported are the ones that use the ATmega328 chip. Similarly to HaikuVM, the occam-pi bytecodes are stored in flash memory alongside the virtual machine. However, unlike HaikuVM, the bytecode can be uploaded separately. Another similarity ocamm-pi has with HaikuVM is the static program analysis that allows it to eliminate dead-code and generate compact programs. This process is not only performed on user-generated code but also on ocamm-pi libraries. Ocamm-pi has a rich set of runtime libraries that provide functions for interacting with Arduino features such as the serial port, PWM and TWI. Most of these libraries are entirely implemented in occam-pi. This is possible thanks to interrupts and memory being accessible from occam-pi code, allowing the development of low-level libraries directly in occam-pi. However, handling interrupts using occam-pi code has a performance cost that limits the amount of information that can be processed. For example, handling serial communication in occam-pi can only process characters at a baud rate of at most 300 bps. Regarding performance, the execution of bytecodes has been reported be 100–1000 times slower than the execution of native code.

Splish [8] is an interesting project because instead of providing only a virtual machine it also provides a visual programming environment, much like PE. All the instructions and programming constructs are represented as icons that can be interconnected to define the program flow. The programs built using this visual environment are then compiled into an object code for a stack virtual machine designed specifically for this language. Uploading the compiled programs into the Arduino board is done via USB. The Splish firmware includes a monitor program that is in charge of the communication between the board and the computer; it listens to the Serial port for commands to execute and periodically sends back status information. This allows the computer to monitor the state of the Arduino pins and the execution of the programs. It is designed with debugging facilities in mind, even if that has a negative impact on the performance. If the Arduino board is connected to the PC, it can run programs in “debug mode”, allowing step by step execution.

PyMite [9] (also known as python-on-a-chip) is a Python interpreter for 8-bit and larger microcontrollers. It can execute a subset of Python bytecodes and it supports almost all of Python’s most important data types (such as 32-bit signed integers, Strings, Tuples, Lists, and Dictionaries) and some advanced features such as generators, classes, and decorators. It allows writing native code by marking a Python function with a special keyword and writing the C code in the function’s documentation string, thus making it easy to develop low level libraries. It supports several platforms, but since it requires at least 64 KB of program memory and 4 KB of RAM, Arduino boards smaller than the MEGA are not supported.

The Scheme programming language has several implementations designed for small microcontroller based embedded systems. Two of the most interesting ones are Microscheme [10] and PICOBIT [11], which are very different in their approach, even though they both are implementations of the same programming language. Microscheme targets the 8-bit ATmega chips used by most Arduino boards, while PICOBIT targets the Microchip PIC18 family of microcontrollers. Microscheme differs from PICOBIT in that it uses direct compilation instead of a virtual machine. Its compiler, written in C, generates AVR assembly code which is in turn assembled and uploaded to the board using the avr-gcc/avrdude toolchain. PICOBIT instead provides a Scheme virtual machine written in portable C that, although being currently implemented for the PIC18 microcontrollers, could be ported to any platform that has a C compiler. Another characteristic worth mentioning of the PICOBIT approach is that it does not only provide a custom Scheme compiler and virtual machine but also a custom C compiler designed specifically for developing virtual machines. This C compiler takes advantage of the patterns commonly found in the implementation of virtual machines and it performs a set of optimizations that result in a significant reduction of the generated code. Both implementations support different subsets of Scheme.

3 Design Principles

The main goal of this project is to provide a tool that a visual programming environment such as PE could use to compile and run its programs.

Given that PE has an educational purpose, Uzi was designed based on the following principles:

- Simplicity: It should be easy to reason about the virtual machine and how it does its job.
- Abstraction: It is the responsibility of the Uzi language to provide high-level functions that hide away some of the details regarding both beginner and advanced microcontroller concepts (such as timers, interruptions, concurrency, pin modes, and such). These concepts can later be introduced at a pace compatible with the needs of the user.
- Monitoring: It should be possible to monitor the state of the board while it is connected to the computer.
- Autonomy: The programs must be able to run without a computer connected to the board.
- Debugging: Uzi must provide mechanisms for error handling and step by step execution of the code. Without debugging tools, the process of fixing bugs can be frustrating for an inexperienced user.

4 Implementation

In order to simplify the translation of PE scripts to Uzi programs the execution model of the Uzi virtual machine was designed to be as similar as possible to the Etoys model. Like Etoys projects, an Uzi program can include several scripts that are executed concurrently. Each script runs forever in an implicit loop and its execution rate can be configured independently. This model simplifies the needed code to express concurrent tasks, as can be seen in the example section of the present paper.

Some of the Uzi tools reside on the computer while others run directly on the Arduino. The computer counts with all the necessary components to parse, compile, and transmit the programs to the Arduino board through the serial port. All these programs were developed using Squeak, an open source version of Smalltalk. The Arduino, on the other hand, includes all the software required to execute Uzi programs. These tools were written in C++ (Fig. 1).

The UziParser is responsible for parsing a string written in Uzi syntax and generating a parse tree. It was implemented using PetitParser [12], a parsing framework that allows you to define parsers using Smalltalk code. Although the Uzi syntax is heavily inspired by Smalltalk, it should not be confused with Smalltalk code. Uzi does not follow any of the Smalltalk semantics. It does not support objects nor late binding. It is a domain specific language which main purpose is to run PE scripts inside the Arduino board and it was designed to make the translation process as simple as possible.

An example script written in the Uzi language can be seen below. This small program blinks the LED on pin 13:

```
#blink13 ticking 1 / s [toggle: 13]
```

The UziCompiler is responsible for traversing the parse tree and generating a compiled program containing the bytecodes that the Uzi virtual machine will execute.

The UziEncoder is in charge of serializing the program to a custom binary format designed for Uzi. This format is designed to be as compact as possible

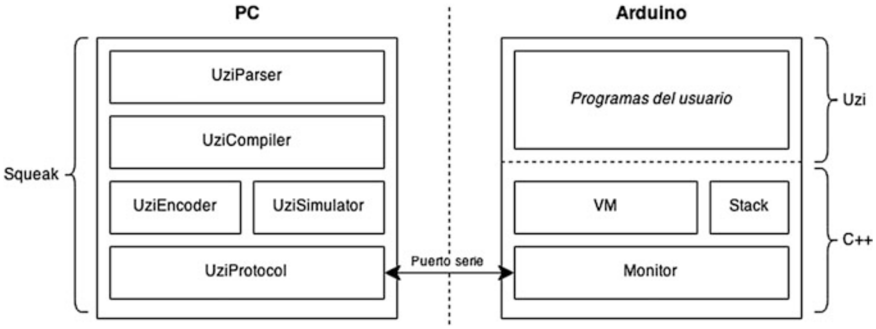


Fig. 1 Uzi architecture

because it will not only be used to transmit the program to the arduino but also to store it in the EEPROM, which has very limited space.

The UziSimulator is a Smalltalk implementation of the Uzi virtual machine. This tool allows us to run on the computer the exact same process that the Arduino will execute. This is currently useful to verify the implementation of new functionality before making the change in the actual virtual machine that will be uploaded to the board. In the future, the UziSimulator might also be used to add debugging features such as step by step execution.

The UziProtocol is the last tool on the PC side. It is used by the other components to communicate with the arduino. It can either send entire programs or specific commands that the arduino will execute. It also listens for arduino state updates. All the IO primitives implemented on the UziSimulator, for example, use the UziProtocol to actually perform the operation on the board.

On the arduino side, Uzi is installed as a firmware that contains both the Uzi virtual machine and also a small Monitor program that communicates with the UziProtocol through the serial port. The Monitor acts as a bridge between the virtual machine and the development tools. It listens on the serial port waiting for commands to execute and periodically sends data back to the computer. The commands that the Monitor understands include IO operations, executing a specific program, and storing a program in the EEPROM memory. The data that the Monitor sends includes the state of the pins and the state of the virtual machine (global variables, instruction pointer, stack, and current script).

The VM class is responsible for executing Uzi programs. It requires, essentially, two attributes: the instruction pointer (IP), an integer that refers to the next instruction to be executed; and a pointer to the stack. In each tick, the VM iterates over the entire list of scripts. For each script the VM knows the time it was last executed and its ticking rate. If the time since it was executed exceeds its ticking rate, the VM executes the script. Executing a script involves resetting the IP and executing each of the script's bytecodes one by one. The execution of a script must leave the stack exactly as it was before its execution started. The bytecode execution is handled by a simple switch statement. Since, as mentioned before, the Uzi compiler privileges small code size over execution speed, the Uzi instruction set was designed to use as little space as possible. Each instruction occupies one byte, where the most significant four bits are used to represent its operation code and the least significant 4 bits are used to specify its operand. Since 4 bits can only address a maximum of 16 values, a special instruction is used to extend a specific operation by using the next byte as its operand. The value 0xFF is used to mark the end of a script. Since only 4 bits are used to represent an operation code, the instruction set only includes the most common operations, such as handling the stack, accessing pins, calling primitives, and starting/stopping scripts. Other operations (such as arithmetic or logical) are implemented as primitives.

The stack has a fixed size of 100 elements. In case of stack overflow, the VM will stop execution immediately. The invalid state will be stored and transmitted by the Monitor to the host PC (if connected).

5 Example

The following example, although admittedly simple, is useful to show the differences between code written in the simplified C environment provided by Arduino (which we will call Arduino code), scripts built using the PE visual interface, and scripts written in the Uzi programming language.

This program performs four independent tasks:

1. It blinks a LED once per second (BLINK13).
2. It blinks another LED twice per second (BLINK12).
3. It turns on a third LED when a button is pressed (BUTTON).
4. It controls the brightness of a fourth LED with a potentiometer (DIMMER).

These simple tasks are performed concurrently, which is something difficult to express in the Arduino code. As it can be seen in the example below the Arduino code mixes the statements that perform the tasks with the code required to schedule them at the correct intervals. This makes the code's intention less obvious and, thus, harder to read and modify.

The Arduino conceptual model for the pins represents another issue. In order to read or write, it differentiates analog and digital pins, forcing the user to use different functions for different types of pins. The abstraction is not event correct: the function that "writes" a PWM wave is called analogWrite() even though it does not generate an analog wave and is not related to analog pins or the analogRead() function in any way [13]. Additionally, each pin can either be in one of two modes, which the user must explicitly specify: INPUT for reading, and OUTPUT for writing. A simpler model could restrict the operations that can be performed on a pin to simply "write" and "read", handling each specific case without exposing the details to the user. This model has its drawback but it would be simpler to understand for a beginner than the Arduino functions. Moreover, while digitalWrite () and digitalRead() functions work on the same range (either 0 or 1), analogWrite() accepts a value from 0 to 255 and analogRead() returns a value from 0 to 1023. This small difference forces the user to transform from one scale to the other when trying to use the input from an analog pin to output a PWM signal, as can be seen in the Arduino example code. Failing to do this can lead to incorrect behavior, which is difficult to debug for an inexperienced user.

Additionally, since you can't read the value of a pin configured as OUTPUT (without accessing the registers directly, at least), in order to blink the LEDs, the user is forced to store the state of the pins in a variable. This extra code adds complexity to the solution.

Handling each LED blink rate also requires extra code. Using the delay() function, which blocks the processor for a given amount of time, as it is a common practice in Arduino examples, is not allowed here because it would disrupt the execution of the other tasks (the Arduino board has only one microcontroller). Instead, the user is forced to call the millis() function and check on every tick if it is time to blink each led.

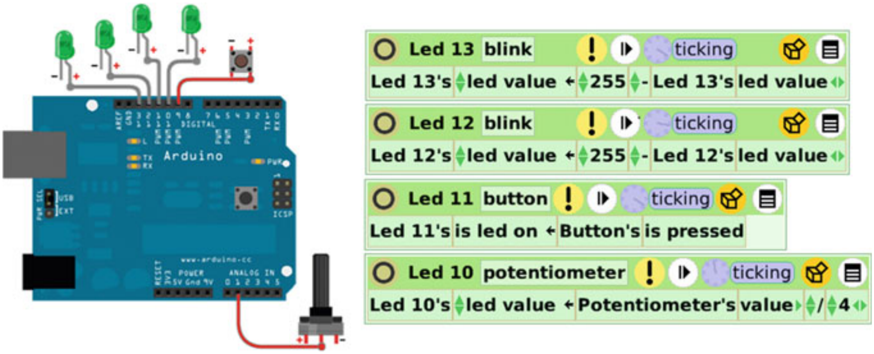
All these issues with the Arduino code greatly increases the complexity of an otherwise simple project.

```
boolean leds[] = { false, false };
unsigned long last = 0;

void setup() {
 for (int pin = 10; pin < 14; pin++)
   pinMode(pin, OUTPUT);
 pinMode(9, INPUT);
 pinMode(A1, INPUT);
}
 void loop() {
  unsigned long now = millis();
  if (now != last) {
    if (now % 1000 == 0) toggle(13); // BLINK13
    if (now % 500 == 0) toggle(12);  // BLINK12
    last = now;
  }
  digitalWrite(11, digitalRead(9)); // BUTTON
  analogWrite(10, analogRead(A1) / 4);  // DIMMER
 }

 void toggle(int pin) {
  leds[pin - 12] = !leds[pin - 12];
  digitalWrite(pin, leds[pin - 12] ? HIGH : LOW);
 }
```

In PE this same example is very different due to the fact that PE is a completely visual programming environment. First, the user needs to indicate which type of device is connected to each pin by clicking and dragging on icons. Then the user has to build each script by, again, clicking and dragging the different instructions. Each script belongs to an object and it runs concurrently with all the others. The concurrency is automatically handled by the PE scheduler, which simplifies describing the execution of concurrent tasks (Fig. 2). Although it cannot be seen in the figure, each script is configured to run at different rates: 1/s. for the first, 2/s for the second, and 100/s for the third and fourth scripts. Such configuration is much simpler to set up in PE than in the Arduino code. Each task is encapsulated into its own script, which simplifies reading and understanding the code. The visual interface presented by PE is easier for beginners to understand because it makes syntax errors impossible and it exposes the user to an object oriented API in which each graphical object represents a real object that the user can manipulate directly. Although the user does not have to specify each pin mode, it does have to tell PE which devices are connected to each pin, but doing it by clicking and dragging



Led 13 blink ticking
Led 13's led value ← 255 - Led 13's led value

Led 12 blink ticking
Led 12's led value ← 255 - Led 12's led value

Led 11 button ticking
Led 11's is led on ← Button's is pressed

Led 10 potentiometer ticking
Led 10's led value ← Potentiometer's value / 4

Fig. 2 Graphical representation of the arduino inside PE and its corresponding scripts

devices into their corresponding pins feels much more natural and intuitive than calling a function.

Finally, the Uzi program is the smallest of the three, with only four lines of code describing four scripts. Each script can be configured with its own ticking rate and the Uzi virtual machine will take care of executing it at the desired interval. If the user does not specify a ticking rate (as it is the case with the “button” and “dimmer” scripts, then the virtual machine will execute them on every tick). It is no longer necessary to remember the state of each pin in order to blink the LEDs, because Uzi handles it automatically when calling the “toggle:” primitive. There is no distinction between analog and digital pins, the only operation available is “write:value:” and “read:” (apart from others that can be built upon these two, such as “toggle:”) and both accept values in the 0 to 1 range. This can be seen in lines 3 and 4, where the scripts have essentially the same statements but with different parameters. And finally, the user is not forced to specify each pin mode explicitly; Uzi configures the pins automatically.

```
#blink13 ticking 1 / s [toggle: D13]
#blink12 ticking 2 / s [toggle: D12]
#button ticking [write: D11 value: (read: D9)]
#dimmer ticking [write: D10 value: (read: A1)]
```

6 Limitations

Some of the design decisions that were taken during the implementation of Uzi resulted in limitations, performance being the most important. Using a virtual machine makes it nearly impossible to obtain the same performance that can be obtained using native code. Although no benchmarks have been run yet, we expect the performance to be at least 100x slower. For most of the programs we expect the users to write using PE this might not be a problem, but for others this might impose a clear limitation. One of the solutions that is being considered is to automatically generate, apart from the Uzi bytecodes, a C++ program that could be uploaded to the board using the arduino software. This would allow the maximum efficiency for the cases when it is needed while maintaining the benefits of the virtual machine approach.

The small amount of memory available on Arduino boards poses a whole set of limitations. Currently, the EEPROM is being used exclusively for program storage, which means that the user cannot use it to store values from its programs. Ideally, the Monitor, VM and user programs would all be stored in Flash memory (which is much larger), but this is not implemented yet. Until then, no strings or arrays are supported because they occupy too much space.

The current Uzi implementation does not allow dynamic memory allocation. This design decision has several advantages, such as making the implementation of a garbage collector unnecessary or allowing static analysis to determine how much memory the program will need at compile time. However, it also restricts the type of programs that can be written using Uzi.

7 Future Work

Uzi is still a work in progress. Although most of its design is finished, only a small subset of all primitives is currently implemented, which allows to write only simple programs like the one described above. Once the implementation becomes stable, it will be integrated with PE so that visually scripted Etoys projects can be translated to Uzi bytecodes.

The Uzi language also requires better tooling. Although debugging is one of the project guiding principles, no debugger has been implemented yet. The development of an integrated development environment is planned for the future.

Even though Uzi is currently designed with a special focus on PE, it is of interest for the authors to evaluate its capabilities as an intermediate language in which different programming models could be implemented.

Finally, since the Uzi virtual machine is small and simple, porting the Uzi virtual machine to other educational robotics platforms (such as Lego Mindstorms Nxt or even PIC microcontrollers) is also of interest.

8 Conclusion

The design and implementation of Uzi, a virtual machine for Arduino, was described.

This virtual machine solves a specific problem encountered while using PE to teach robotics to high school students.

The advantages of Uzi over the traditional Arduino tools were exemplified by writing the same program in three different programming languages: the simplified C provided by Arduino, PE, and Uzi.

Given the advantages of Uzi over the traditional Arduino tools, its use for educational purposes is highly encouraged.

References

1. Freudenberg, B., Ohshima, Y., Wallace, S.: Etoys for one laptop per child. In: 7th International Conference on Creating, Connecting and Collaborating through Computing—C5 2009, Kyoto, pp. 57–64 (2009)
2. Citilab: Scratch for Arduino (2015). http://s4a.cat/
3. Rahul, R., Whitchurch, A., Rao, M.: An open source graphical robot programming environment in introductory programming curriculum for undergraduates. In: 2014 IEEE International Conference on MOOCs, Innovation and Technology in Education, IEEE MITE 2014, Patiala, pp. 96–100 (2014)
4. Bob Genom: HaikuVM: a small JAVA VM for microcontrollers (2014). http://haiku-vm.sourceforge.net/
5. Rao, A.: The application of LeJOS, Lego Mindstorms robotics, in an LMS environment to teach children Java programming and technology at an early age. In: 5th IEEE Integrated STEM Education Conference, ISEC 2015, pp. 121–122(2015)
6. Elizabeth, M., Hull, C.: Occam-a programming language for multiprocessor systems. Comput. Lang. **12**(1), 27–37 (1987)
7. Jacobsen, C.L., Jadud, M.C., Kilic, O., Sampson, A.T.: Concurrent event-driven programming in occam-π for the Arduino. Concurr. Syst. Eng. Ser. **68**, 177–193 (2011)
8. Kato, Y.: Splish: a visual programming environment for arduino to accelerate physical computing experiences. In: 8th International Conference on Creating, Connecting and Collaborating through Computing, C5 2010, La Jolla, CA, pp. 3–10 (2010)
9. Python (2014). https://wiki.python.org/moin/PyMite
10. Suchocki, R., Kalvala, S.: Microscheme: functional programming for the Arduino. In: Scheme and Functional Programming Workshop, Washington, D.C., pp. 21–29(2014)
11. St-Amour, V., Feeley, M.: PICOBIT: a compact scheme system for microcontrollers. In: 21st International Symposium on Implementation and Application of Functional Languages, IFL 2009, South Orange, NJ, pp. 1–17 (2010)
12. Bergel, A., et al.: PetitParser: Building Modular Parsers. In: Deep into Pharo, pp. 375–410 (2013)
13. Arduino—analogWrite() (2015). https://www.arduino.cc/en/Reference/AnalogWrite

Model-Based Design of a Competition Car

Richard Balogh and Marek Lászlo

Abstract The paper shows how students used the modeling and simulation capabilities of the Matlab/Simulink to improve the control design of their winning FEIminetors car for the worldwide known Freescale Cup competition. Creating and simulating the model gives us a better understanding of the processes and almost bug-less transfer of the code to the embedded processor. Model was used also for the first estimation of a device controller. We also summarize our experiences with the competition organization.

Keywords Freescale cup · Model-based design · Automotive design · Student competition

1 Introduction

The Freescale Cup [1] is a global engineering multilevel competition where student teams build, program and race an intelligent autonomous model car around a track. The fastest car to complete the track without going off the track wins the race. The competition aims to deepen student's knowledge about the embedded control systems design.

During the design phase students must tackle several Science, Technology, Engineering, and Math (STEM) related issues such as embedded microcontroller programming, closed loop control, modeling and implementation, as well as overall vehicle dynamics (physics). Soft skills are also trained through team collaboration, communication and project management [2].

Detailed car design documentation is an important part of the registration process, but unfortunately, it is not published anywhere and remains unknown for other teams. Newcomers have to search for some rarely published solutions or have to start from the scratch each year. In previous years some papers were published focusing on

---

R. Balogh (✉) · M. Lászlo
Slovak University of Technology in Bratislava, Bratislava, Slovakia
e-mail: richard.balogh@stuba.sk

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_20

specific aspect of the car design, e.g. the interface board design [3], or the line detection algorithm [4, 5].

Instead of detailed algorithm description, in this paper we try to show how to use a model based design instead of ad-hoc based and intuitive approaches for successful implementation of the controller algorithm.

Brief introduction into the competition rules is sketched in the Sect. 2. In the Sect. 3 we show the electronic differential design, creation of the whole car model and the controller. In the last section we summarize our experiences and recommendations.

2 The Freescale Cup Competition Rules

The Smart Car race was originally conceived in 2003 in collaboration with Hangyang University in South Korea and global company Freescale Semiconductor to increase student exposure to cutting edge industry tools [2]. For the first time, at the Hanyang University, they hosted 80 teams of students. Later the competition significantly succeed in China, where the number of teams quickly reached few hundreds—in 2008, China alone hosted over 1,800 teams from over 600 universities. Since that time the Freescale Cup has spread from Asia to both Latin and North Americas and finally also to the Europe, impacting more than 100 schools and 15,000 students a year [6]. The competition is multi-level, in Europe it consists of few regional qualification tournaments, one EMEA regional championship and finally the world-wide championship. As an example, the total number of 75 students from 25 teams representing their respective universities from 11 European countries raced their cars on the 2014 EMEA region Freescale Cup track at Fraunhofer IIS in Erlangen, Germany. The 180 sq/m racetrack consisted of speed bumps, intersections, hills and chicane curves (Fig. 1).

The spirit of the game is that students demonstrate excellent hardware integration and superior programming.

Competition is for teams of undergraduate students (2–4) from technical universities.

Vehicle must complete a full lap and pass the start/stop line to be recorded and registered. The car must also stop autonomously within 3 m, otherwise it is penalized. Team has 3 attempts to achieve a full lap. The time of the first successful lap is the time recorded for the team. This is also the most significant change since the origins of the competition, where full three attempts were allowed and the learning strategies were supported. Unfortunately, the learning by trial is now not supported. If any part of one or more wheels leave the race surface, the car is considered as failed and the time is not recorded. Absolutely no modifications of the car are allowed after the unknown track is revealed.

Competition car: Organizers of the competition provide the competition kit consisting of the car chassis with motors and electronics (processor, interface board) with

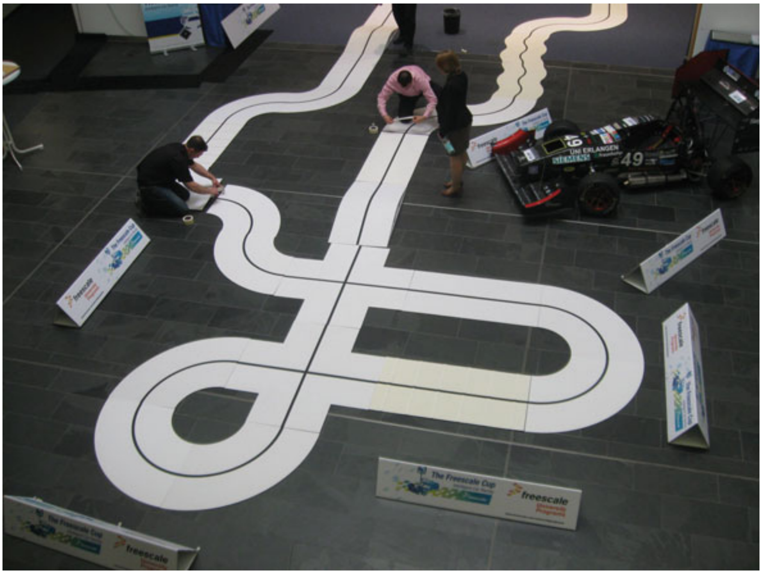


Fig. 1 Building the race track in Erlangen 2014

the battery. The foundation of the car is a model vehicle chassis, transmission, DC motor and servo steering. For the construction, the original and unaltered car chassis must be used. This is especially focused on mechanical parts (tires, dc motors and their transmissions). Footprint of the frame and the distance between wheels may not be altered. Other changes are allowed. The car should be obtained by the students themselves, but due to the high costs, our approach is to support the team with the car.

Even the car is offered with the controller based on the Freedom KL-25Z board with the car-specific shield containing all the necessary interfaces (H-bridge, servo control and camera interface), teams are allowed to design their own electronics, assuming they follow some rules. Only single processor, a Freescale 32-bit MCU must be used. No auxiliary processor or other programmable device is allowed. No DC-DC boost circuits may be used to power drive or steering motors and the total capacity of all capacitors should not exceed 2000 uF. There is an unlimited number of sensors, but Freescale products must be used whenever possible. The base components, chassis, electronics, steering, and drive train are shown in Fig. 2.

Parameters of the Racing Track: The track layout is not known to the challengers until race day. Each year changes are made to the tracks which contain several elements of difficulty including hills, hairpin turns, S-curves, and high speed straight-aways. Width of the racing track is 60 cm, its color is matte white, with a continuous black line (2.5 cm wide) in the middle as the pilot line. Rules also specify the minimum bending radius, intersection angle and slope of the track (see Fig. 3 for an example of racing track).

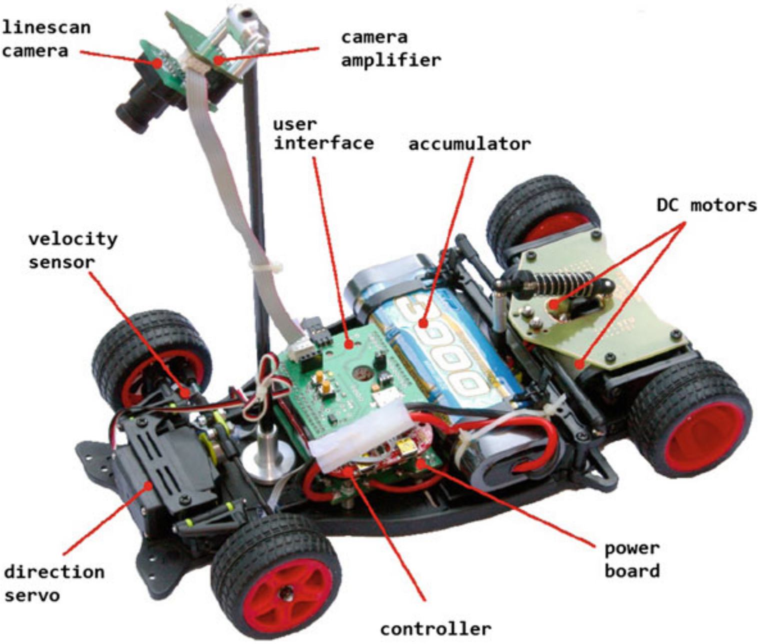


Fig. 2 The car components and modules

Fig. 3 FEI-minetors team (from *left* Norbert Gál, Marek Lászlo and Andrej Lenčucha) at the Freescale Cup Competition with their car

3 Modelling

Winners of the 2014 Freescale Cup EMEA competition, the FEI-minetors team from the Slovak University of Technology in Bratislava were not a newcomers in this competition. During the initial testing and programming of the car, we recognized the need for better understanding of various parameters and car properties. For successful design it was necessary to understand not only the basic physics beyond the car and the properties of the proposed controller, but we also need to know how certain parameter influences another and which of them are the most important factors.

As the students of the STU in Bratislava are using the Matlab/Simulink during the standard academic courses, it was a natural choice to use it also for modelling the car for the Freescale Cup. As an illustration, we will show two important models created for the competition.

3.1 Electronic Differential

The car contains two DC motors in the Ackermann steering geometry chassis, so it was necessary to implement so called electronic differential to safely drive all the curves in high speeds. Its function is to modify the speed of inner and outer wheels according the steering angle. For better understanding of its function and for easier implementation of the function in the embedded microcontroller, model of the steering geometry was created. It includes lot of parameters, starting from geometry (dimensions of the wheels, radius, length of the axis), including motor properties (speed, torque).

Resulted differential model was simulated in Matlab and results were displayed in the 3D chart of the Speed/Torque and Steering Angle for both left and right wheels (see Fig. 5). Excerpt of the code used for simulation is on Fig. 4. It was later implemented in the C programming language also in the embedded microcontroller. Later the model and its parameters were modified based on real tests and empiric knowledge. As an example, the value of the speed exponentiality c was adjusted.

3.2 Controller Design

For modelling, simulations and testing much more complicated model was created (see Fig. 7). In Simulink, it was quite easy to start with modelling subsystems (car geometry, DC motor, controller, etc.) and then integrate them into the one complex layered model. We started with the standard text-book model [7] of the DC motor with some measured and some more empiric parameters (see Fig. 6). Model combines standard electric and mechanic dynamic model into the single one. Even in this early stage, saturation and non-linearity were included into the model.

```
torque     = 400;    % Torque setpoint
wheel_base = 1.6;    % From rear to front wheel   [m]
track      = 1.24;   % Between wheels             [m]

% car driving radius
R = wheel_base/tand(steering_angle);

%  angular speed of car
omegav = v(index1)/R;

%  ideal rear wheels peripheral speed
vR = omegav*(R-(track/2));
vL = omegav*(R+(track/2));

t_ratio_r = 1-(v(index1)/vR);
t_ratio_l = 1-(v(index1)/vL);

% calculate differential values
t_diff_r = torque/2 * t_ratio_r * (exp(c*v(i1)));
t_diff_l = torque/2 * t_ratio_l * (exp(c*v(i1)));

% calculate new rear wheel torque
tR = torque/2 + t_diff_r * gain;
tL = torque/2 + t_diff_l * gain;
```

Fig. 4 Part of the code for calculation of the torques

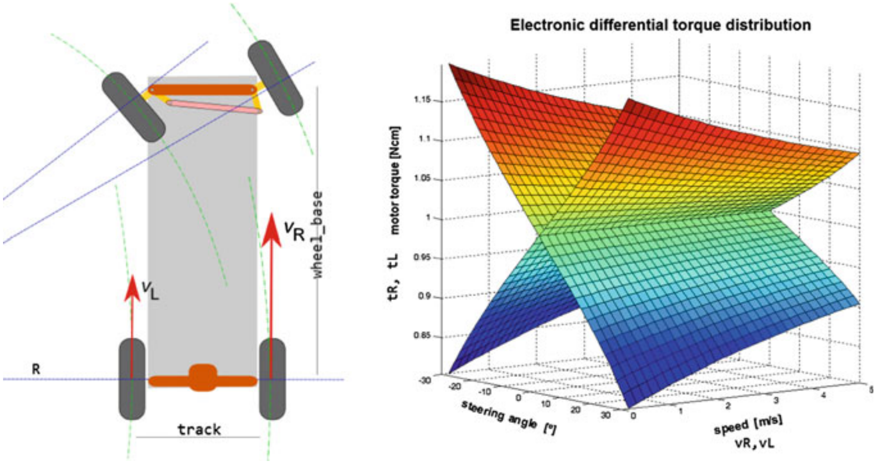


Fig. 5 Ackerman drive model and 3D chart of the electronic differential. Two planes represents the respective torques for the rear left and right wheels. See the Fig. 4 for the code

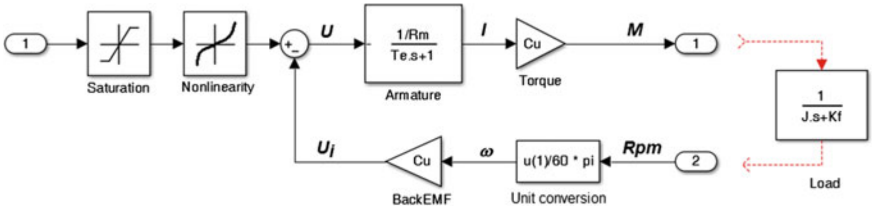


Fig. 6 Model of one of the DC motors

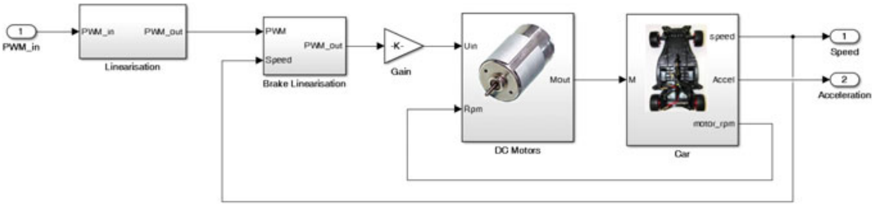


Fig. 7 Model of the car for the speed controller design

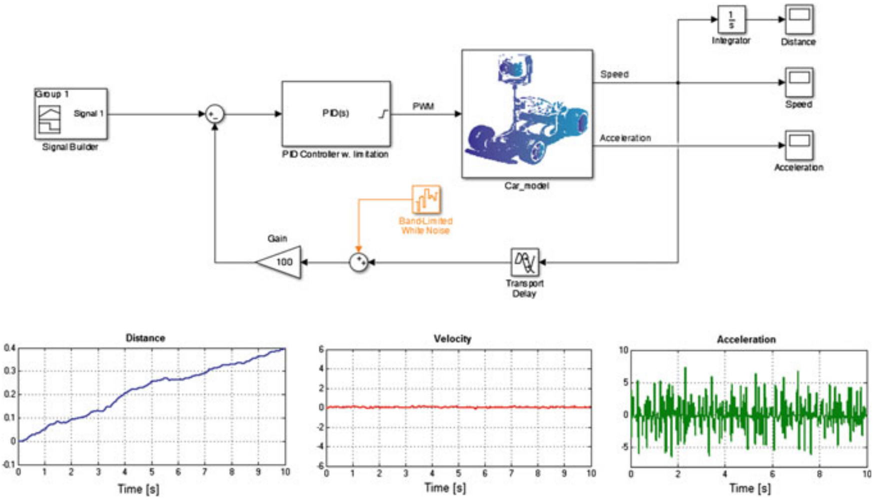


Fig. 8 Study of the noise influence on the proper controller operation

Later we added also model of the car chassis including its weight, dimensions—see Fig. 7. Model of the DC motor from the previous step is the essential part of it. Most of the car parameters were obtained by the measurement (mechanical dimensions) and from experiments (conversion factors). From the beginning, it shows that considering model non-linearities is crucial for good correspondence of the model with reality. In each step, the parameters were adjusted according the measurements to obtain real results.

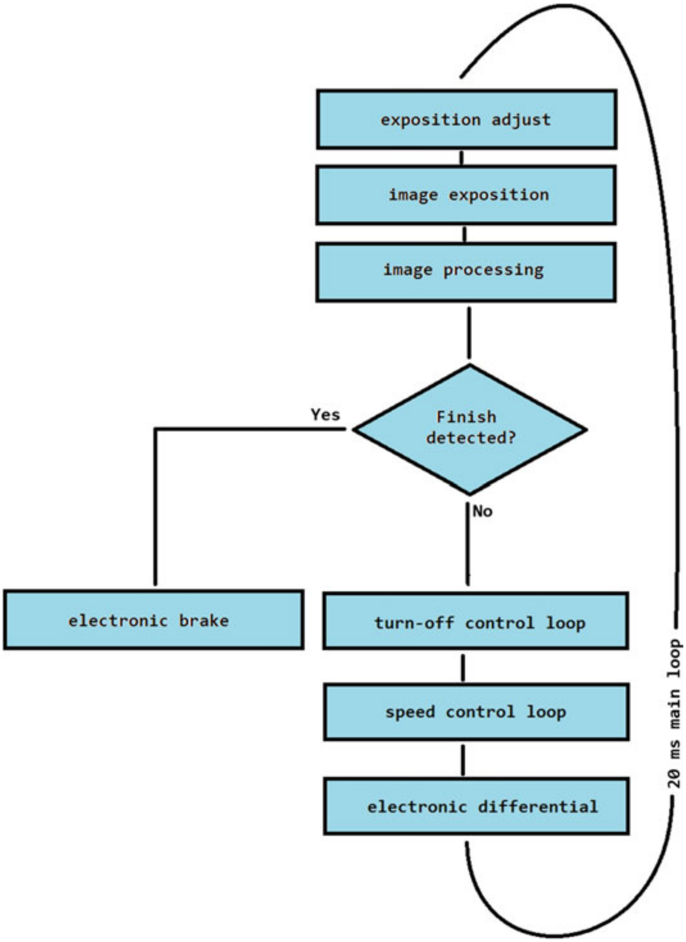


Fig. 9 Main control algorithm

After the subsystems were tested and compared with the real system, everything was combined into the one complex model (see Fig. 8). The main controller of the system was tested and parameters were modified many times based on results of the simulations. Numeric values were obtained with a multiple iterative processes from the first estimation and later adjustments based on experiments and their fitting to the model. One of the most important observations were the current spikes appearing almost everywhere and the final design of the power stage electronics takes this into the account. Later, based on the real measurements and observations, further modifications of the model was included. Finally, the controller code was almost without changes transferred to the microcontroller. In the future, we plan to use the Matlab embedded coder [8, 9] for the Freescale Freedom Board [10] platform, which we are just testing. Advantages of such hardware-in-the-loop modelling and testing were described many times, see e.g. [11].

Later we studied the influence of non-linearities in the system. Controllers behaviours without and with non-linear parts in the model were compared. For speed profiling we needed to use PID controller which had bad quality without linearization. We were measuring steady state speed in correspondence to selected input in steps. Found non-linearity was than corrected.

The main loop runs with the sampling period of 20 ms (see Fig. 9). Control signal is based on the pre-processed camera output. Within the cascade of the two controllers, upper level controls the turn-off and its control signal is based on the deviation of the line from the center position. Bottom level controller is based on the previous controller output error. Its PID algorithm adjusts the overall speed of the vehicle. Zero controller error means that the line is perfectly centered and the car can use its full speed. The largest the deviation of the black line from the central position is, the slowest speed is used to drive the car, providing the enough time to turn-off the vehicle and find the central position again. Output signals are then processed in the module of the electronic differential where the actual speeds of both wheels are calculated. Result is then feed to the appropriate DC motor in the form of PWM signal. The model was also used for testing the influence of the noise induced on various levels (see Fig. 8) and its results were used when designing the shielded cables from the camera sensor.

4 Conclusion

Organization: In China, the Ministry of Education has embraced the event making it standardized curricula across several of the leading universities [2]. Also at some other universities the competition car design was successfully included into their curricula (see e.g. [12]) but this was not our case.

At the STU the students prepare for the competition in their free time only. We attempted to involve the students within their compulsory team-projects, but those were not so motivated and successful. Personal motivation of the students include the deep understanding of the new technology, real problems and possibility to compare with other teams.

Key to the success in our case was the ability and passion to spent many hours with testing and redesigning the car again and again. The task seems to be an easy, but for students, especially in their first years at the university this is not true. They have to fight with a new software packages, their installation, mastering the development cycle, and with all the hardware issues at the same time. Especially when the team choose to design also their proprietary hardware, the time constraints were crucial. Otherwise, probably the main challenge was to implement the image processing routines into the embedded microcontroller.

Experiences: Creating and simulating the model gives us (a) better understanding of the processes and (b) almost bug-less transfer of the code to the embedded processor and (c) first estimation of the controller parameters.

Model is not perfect, but even in this form it is better than nothing. It helped us to inderstand the behaviour of the car, which physical quantities are important more and which are less. This leads to understanding, what can be neglected and what is important. It is a big step over the trial-and-error approach often applied by students.

Model adjusting shows us a lot of non-linearities of the real system, especially limits are important. This is rarely seen in textbooks examples on controller design.

In this paper, we showed just selected parts of the car modelling, as described in [13], which is available on request (in Slovak only). We intentionally didn't mention the modelling of the steering servo, and we also didn't mention the problems with the image sensor and with the line detection in the signal due to the lack of the space available.

Racing is a challenge that virtually every human knows, has no language barriers, and never fails to provide excitement, adrenaline and with it a platform to educate [2].

Acknowledgments Publication of this paper has been supported by the Slovak Research and Development Agency, Grant No. APVV-0772-12.

References

1. The Freescale: The Freescale Cup 2014 EMEA Rules v 1.2 11 (2014). https://community.freescale.com/docs/DOC-94949
2. McLellan, J., Mastronardi, A.: Engaging students: the growing smart car competition. In: 2009 Annual Conference and Exposition, Austin, Texas (2009). https://peer.asee.org/4983
3. Zhicong, S., Xuemei, L., Mei, C., Hongbin, Z.: The design of smart car based on Freescale processor. In: 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering (CCTAE), Jun 12, vol. 2, pp. 508–510. IEEE (2010)
4. Wang, Z., Liu, Y.: Design of road tracing navigation control for smart car use CCD sensor. In: 2010 International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT), April 17 vol. 1, pp. 345–348. IEEE (2010)
5. Xiuquan, W., Xiaoliu, S., Xiaoming, C., Ying, C.: Route identification and direction control of smart car based on CMOS image sensor. In: ISECS International Colloquium on Computing, Communication, Control, and Management, 2008. CCCM'08, vol. 2, pp. 176–179. IEEE (2008)
6. Mc Lellan, J.: History of the Freescale Cup (2016). https://community.freescale.com/docs/DOC-1011
7. Krishnan, R.: Electric Motor Drives: Modeling, Analysis, and Control. Prentice Hall (2001)
8. de Sean, W.: MATLAB and Simulink for Embedded Systems and Robotics. The MathWorks, Inc. (2014). https://www.buffalo.edu/content/www/ccr/support/software-resources/compilers-programming-languages/matlab/_jcr_content/par/download_1/file.res/MathWorks_MATLAB_and_Simulink_for_LEGO.pdf
9. Hahn, C.: Autonom fahren auf unbekannter Rennstrecke. MathWorks untersttzt den Freescale Cup. Matlab Expo Deutschland, MathWorks (2015)

10. Semiconductor, Freescale. FRDM-KL25Z: Freescale Freedom Development Platform for Kinetis KL14. 15/24/25 MCUs. Documentation. http://www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards/freedom-development-platform-for-kinetis-kl14-kl15-kl24-kl25-mcus:FRDM-KL25Z
11. Kelemenová, T., Kelemen, M., Miková, L., Maxim, V., Prada, E., Lipták, T., Menda, F.: Model based design and HIL simulations. Am. J. Mech. Eng. **1**(7), 276–281 (2013). http://pubs.sciepub.com/ajme/1/7/25/
12. Zhao, Z.: The reform and practice of education mode in multi-disciplinary joint graduation design. Procedia Eng. **15**, 4168–4172 (2011). ISSN 1877-7058. http://dx.doi.org/10.1016/j.proeng.2011.08.782 (2011)
13. Lászlo, M.: Navigation and robot control with speed optimalization. Master thesis. Slovak University of Technology in Bratislava (2015). http://www.crzp.sk/crzpopacxe/openURL?crzpID=42928&crzpSigla=stubratislava

Part V

Measuring the Impact of Robotics on Students' Learning

Student-Robot Interactions in Museum Workshops: Learning Activities and Outcomes

Alex Polishuk and Igor Verner

Abstract Our study proposes an approach based on learning through interaction with robots. We investigated how this approach can contribute to the development of systems thinking skills in elementary school students. 346 students of 2nd to 4th grades participated in the workshop “Animal-like robots” conducted at MadaTech. The workshop activities focused on inquiry and design of robot behaviors. We evaluated students’ performance of the workshop tasks and found indications of significant progress in the systems thinking skills.

Keywords Science museum · Workshop · Learning through interaction · Robot behavior · Systems thinking

1 Introduction

Educational robotics facilitates experiences, in which learners acquire knowledge in joyful ways. Playful robotics activities are especially meaningful for elementary education to introduce complex concepts such as system, process, structure and behavior to young students [1]. Elementary schools nowadays are being faced with the challenge of teaching such concepts in the science-technology curriculum [2]. According to the curriculum, the students are expected to learn system concepts and apply them to problems of the natural world, technology, and everyday life.

Studies show that the traditional approach focusing on learning through practice in building robots is attractive not for all students, but for those who have technical skills and motivation for hands-on activities [3]. In recent years, there has been a

---

A. Polishuk
Israel National Museum of Science, Technology and Space (MadaTech), Haifa, Israel
e-mail: alex.polishuk@madatech.org.il

I. Verner (✉)
Faculty of Education in Science and Technology, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: ttrigor@technion.ac.il

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_21

growing tendency to develop and implement alternative “pathways into robotics” that put emphasis on understanding, discussion and cooperation, combine science and art. In this regard, robotics finds more and more use in science museums and centers to bring the benefits of the technology enhanced environments into exhibition and classroom [4, 5]. The literature discusses innovative projects, developed through collaboration with academic researchers, where robots play diverse roles as exhibits, entertaining guides and remote facilitators [4, 6–8].

Technion Center for Robotics and Digital Technology Education in collaboration with MadaTech Gelfand Center for Model Building, Robotics & Communication conduct a long-term research aimed to investigate and develop new strategies of learning through interaction with robots. The rationale behind the interaction-based approach is that learning through interaction with a robot is in line with learning from interactive exhibits, the central and inherent activity in science museums. The study focuses on three directions: (1) Learning through inquiry and creating behaviors of animal-like robots, (2) Public robot theatre performances [4], (3) Science lesson mediated by a humanoid RoboThespian [6].

This paper presents a case study in the first direction that followed up the workshop “Animal-like robots” delivered for elementary school students, the main audience of the MadaTech museum. A theatre performance of animated robots was given to the school groups visiting MadaTech [4], to inspire their participation in the workshop. In the paper we discuss the learning activities, student-robot interactions, and outcomes of the workshop.

2 The Workshop Outline

The workshop was given, depending on school’s choice, in two versions: the 9 h basic version and the 18 h extended version. In our study 97 students participated in the basic workshop and 249 in the extended one. Both workshop versions comprised the learning activities presented below:

- *Introduction to robotics.* The students were exposed to live demonstrations and videos of different types of advanced robots and learned basic robotics concepts.
- *Inquiry into robot behaviors.* Each pair of the students got a pre-programmed “RoboCroc” made of the LEGO WeDo kit (Fig. 1a). The students tested responses of the robot to different “stimuli” (move around, voice, light, and touch) and made conclusions about the robot behaviors.
- *Creating behaviors of RoboCroc.* The students designed and implemented new reactive robot behaviors, using the graphic programming language. They practically tested the robot behaviors and iteratively improved them (Fig. 2a).
- *Exploring the RoboCroc pulley system.* The students inquired different options to connect driving and driven pulleys of the robot. They examined how the speed and direction of rotation of the driven *pulley* depends on the connection.

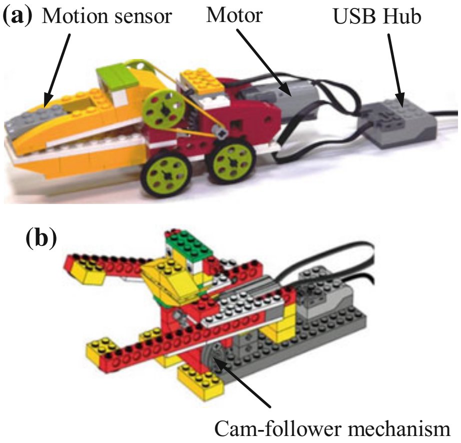


Fig. 1 **a** The RoboCroc. **b** The RoboMonkey

- *Constructing a "RoboDog" and equipping it with a tilt sensor*. The students in pairs built the RoboDogs (presented in Fig. 3) and "trained" (programmed) them to execute commands. The targeting was realized by means of positioning the tilt sensor (Fig. 2b).
- *Creating interactions between RoboCroc and RoboDog*. Pairs performed the assignment working in twos. The students developed scenarios of interactions and implemented them through collaboration (Fig. 2c).

The extended 18 h version of the workshop included additional activities: the students learned how the motor and the control system determine the robot behavior. They also constructed, programmed and tested behaviors of one more robot "Drumming RoboMonkey" (Fig. 1b). This robot had a sound sensor and a cam-follower mechanism to drive RoboMonkey's arms.

3 The Study

Our research aimed to evaluate learning outcomes of 2nd to 4th grade students participated in the robotics workshop. Among more than 2000 participants of the workshop, the research sample included 346 students (grades 2–4) from five different schools in Haifa. In the study the sample was divided into three research groups. The first group was 97 students participated in the 9 h workshop. For this group the consideration focused on the analysis of learning behaviors. The second group of 163 students studied the 18 h workshop. With this group we evaluated and improved the learning activities, as well as tested and validated the worksheets as instructional and research tools. The third group consisted of 86 students who

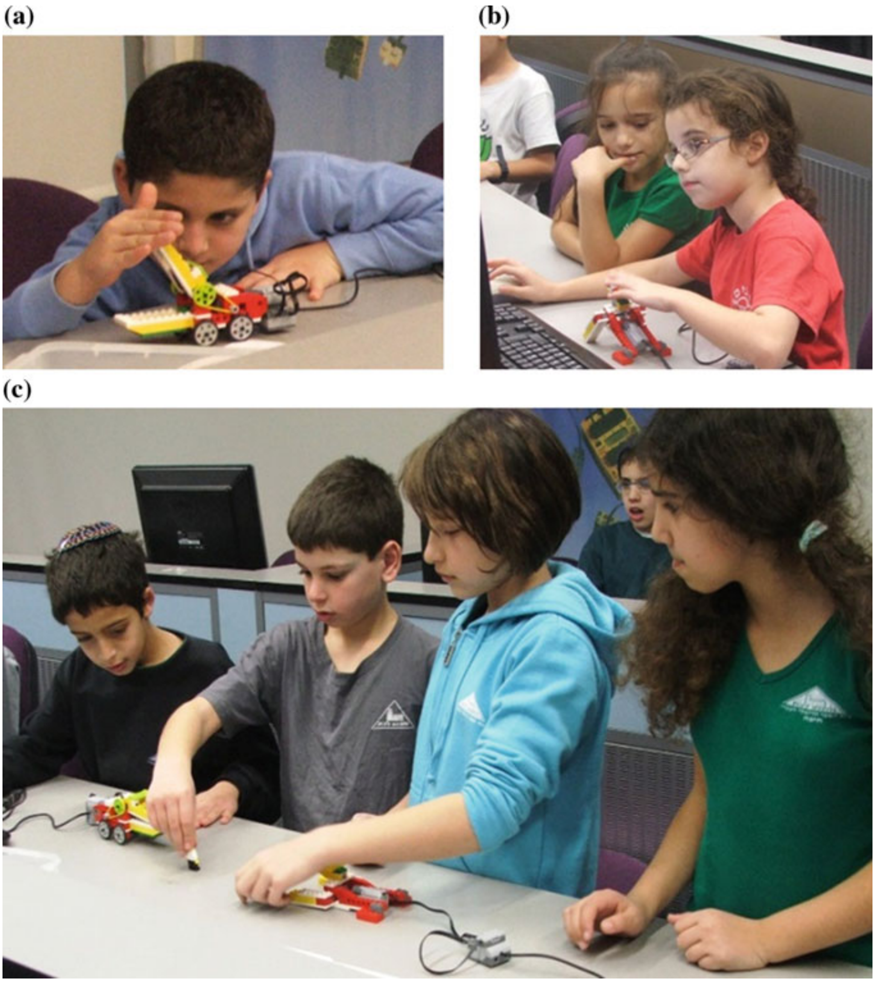
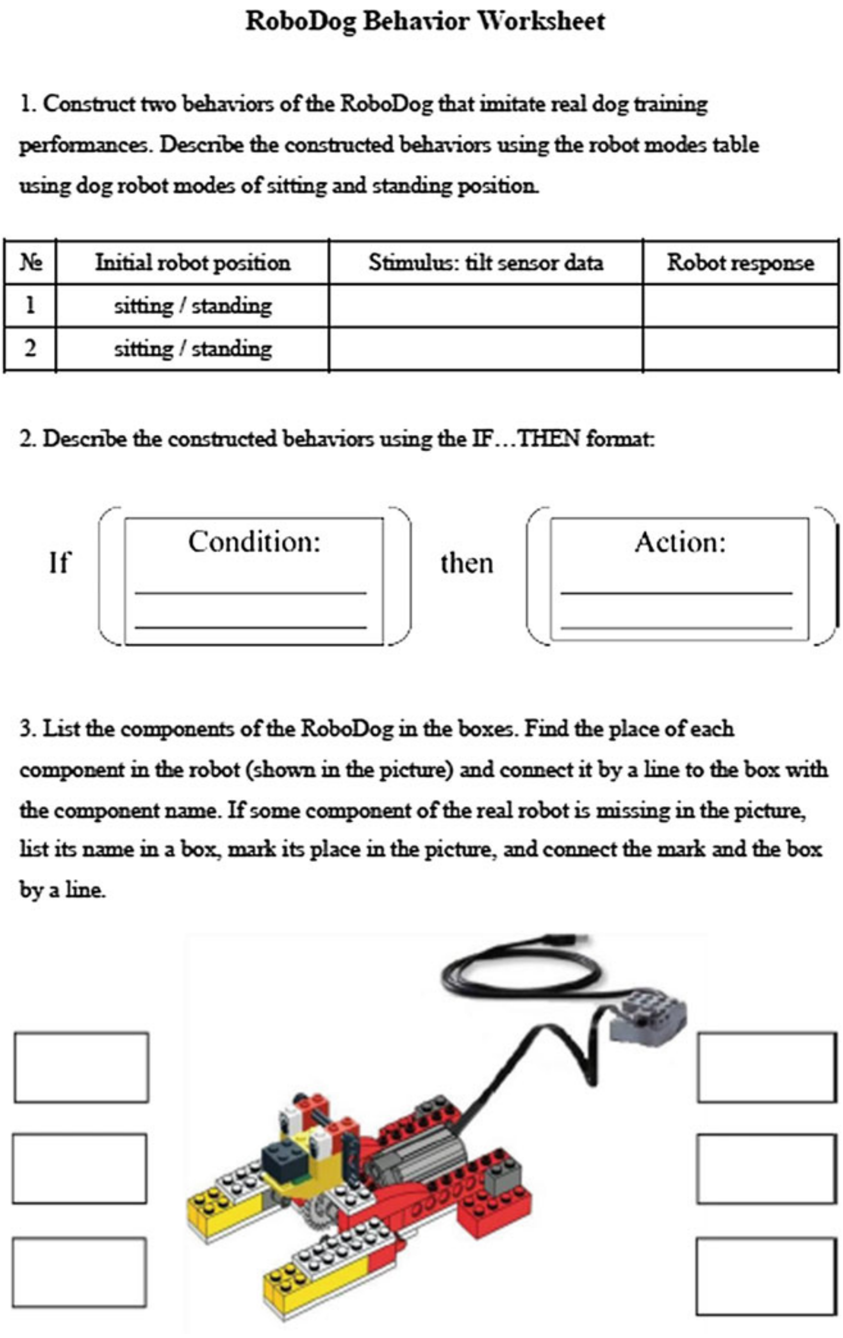


Fig. 2 Testing behaviors: **a**. The RoboCroc; **b**. The RoboDog; **c**. Robot-robot interaction

studied the 18 h workshop, among them 46 s grade and 40 fourth grade students, 51 boys and 35 girls. The study of this group focused on evaluation of the development of systems thinking skills, based on the use of the developed research tools. Here we address the following research question: Are there indications of the development of systems thinking skills in students participated in the robotics workshop? If so, what is the students' progress?

The study applies the model for evaluation of systems thinking skills development in school students, proposed by Richmond [10].



RoboDog Behavior Worksheet

1. Construct two behaviors of the RoboDog that imitate real dog training performances. Describe the constructed behaviors using the robot modes table using dog robot modes of sitting and standing position.

№	Initial robot position	Stimulus: tilt sensor data	Robot response
1	sitting / standing		
2	sitting / standing		

2. Describe the constructed behaviors using the IF…THEN format:

If Condition: then Action:

3. List the components of the RoboDog in the boxes. Find the place of each component in the robot (shown in the picture) and connect it by a line to the box with the component name. If some component of the real robot is missing in the picture, list its name in a box, mark its place in the picture, and connect the mark and the box by a line.

Fig. 3 Worksheet “Design and creating behaviors of RoboDog”

3.1 Richmond's Model of Systems Thinking

Educational literature emphasizes the need for importance of systems thinking and calls for pedagogies that develop systems thinking skills starting from primary school [9]. Richmond [10] proposed such pedagogies to be based on active learning through interaction with and construction of engineering systems. He considered systems thinking as a construct of seven constituent skills: structural thinking, dynamic thinking, generic thinking, operational thinking, scientific thinking, closed-loop thinking, and continuum thinking. Richmond proposed to evaluate the development of systems thinking in students of grades 4–12 by three levels of competence (basic, intermediate and advanced) and characterized these levels for each of the seven skills in general terms.

3.2 Method

The evaluation study was designed to consist of three stages. At the first stage we adapted the Richmond's model for our case, using the inductive method and qualitative analysis of learning behaviors [11]. The data collected by observations and video recording were analyzed through the following steps: students' expressions related to systems thinking, grouping the expressions into patterns, and characterizing each pattern by a feature related to one of the skills defined in the Richmond's model. Using these features, at the second stage we developed five worksheets directed to facilitate systems learning and evaluate its outcomes. The worksheets are discussed in Sect. 3.3. At the third stage we developed scoring rubrics for evaluating the level of student competence, based on the worksheet assignments performance.

3.3 Specific Features of Systems Thinking

Table 1 presents the systems thinking skills in the Richmond's model (1st column), their short interpretations (2nd column) and specific features found at the first stage of the study (3rd column).

3.4 Worksheets as Instructional and Research Tools

Worksheets. The workshop activities were supported by a series of worksheets. The worksheets were used by the students to record the data of experiments in the tasks they performed. The worksheets also were research tools to evaluate the

Table 1 Specific features of systems thinking skills

Skills	Interpretations	Specific features
Structural thinking	Thinking in terms of units of measure or dimensions	Identifying components of the robot structure and their causal relationships. Robot construction and planning
Dynamic thinking	The ability to see and deduce behavior patterns	Testing robot behaviors and their description by flowcharts
Generic thinking	The ability to apprehend the similarities in the feedback-loop relations that generate cycles	Understanding the input-output relations, their representation by the if-then statement, and using the statement for planning reactive behaviors
Operational thinking	Thinking in terms of how things really work	Understanding mechanisms behind robot operation
Scientific thinking	Rigorous hypothesis-testing with modifying only one thing at a time and holding all else constant	Understanding physical parameters of the robot, generating and testing hypotheses about its reactions on stimuli
Closed-loop thinking	The ability to look to the circular cause-effect relations responsible for generating the behavior patterns exhibited by a system	Programming robot behaviors using feedback control
Continuum thinking	The ability to see connections and interdependencies	Identifying, planning and modifying continuous robot movements and interactions between robots

students' progress in systems thinking skills during the workshop. The time gap between the lessons at which the students were assigned to complete the worksheets was 3–4 weeks.

The worksheets were validated by two educational robotics researchers and two science teachers. The list of the worksheets is presented below:

Worksheet 1: Inquiry and design of RoboCroc behaviors. In the 1st assignment of the worksheet the students were asked to document the change of RoboCroc behaviors in response to stimuli. The possible behaviors before enacting the stimuli were given: breathing, attack, and relax (listed in the 2nd column of Table 2). The assignment was to identify and fill in the 3rd column the behavior responses to the stimuli. The students were asked also to identify and record in the table the changes in the robot behavior after ending the stimuli. The 2nd assignment of the worksheet was to propose a new behavior of the RoboCroc and describe it by an "if-then" statement. In the 3rd assignment of the worksheet the students were given a picture of the robot and were asked to list the components of the robot and show their locations in the picture.

Worksheet 2: Design and creating behaviors of RoboDog. The students were asked to document the RoboDog behaviors that they developed. Like in the first

Table 2 Stimulus-response scheme of RoboCroc behaviors

Stimuli	State before enacting the stimulus	State in response to the stimulus
Yes	Breathing	attack
	Attack	retreat-attack
	Relax	retreat-breathing
No	Breathing	breathing
	Attack	retreat-relax
	Relax	relax

worksheet, they described and recorded reactive behaviors of the robot, and identified the robot components (Fig. 3).

Worksheet 3: Design and creating behaviors of "Drumming RoboMonkey. The assignments in this worksheet were similar to that in worksheet 2.

Worksheet 4: Exploration of mechanical transmissions. The students documented results of their inquiry on how the jaw's rotation and body movement are affected by changes of the pulleys' connection.

Worksheet 5: Exploration of cam-follower mechanism. The students documented the effect of changes of the cam-follower mechanism on the drum rhythm and arms synchronization of the RoboMonkey.

The mapping of the seven systems thinking skills addressed in each assignment of the worksheets is presented in Table 3. For instance, each of the first three worksheets consisted of three assignments. The second of them addressed the development of dynamic, generic and closed-loop thinking skills.

Scoring rubrics. The rubrics for evaluation of students' systems thinking skills were developed for all the worksheets. Every rubric relates to a certain category of systems thinking and has criteria for evaluating the assignment performance at three levels (basic, intermediate and advanced). For example, the rubrics related to "Design and creating behaviors of a RoboDog" are presented in Table 4. For each rubric the table presents an assignment name, a criterion, and its specifications for three levels of performance.

Table 3 Mapping of the systems thinking skills

Systems thinking skills	Worksheets 1–3			Worksheet 4	Worksheet 5
	Ass. 1	Ass. 2	Ass. 3		
Structural thinking			☑	☑	
Dynamic thinking		☑		☑	☑
Generic thinking	☑	☑			
Operational thinking				☑	☑
Scientific thinking	☑			☑	☑
Closed-loop thinking		☑			
Continuum thinking	☑				

Table 4 Rubrics for evaluation of the worksheet assignments

Rubric	Skills	Criterion	Task performance levels		
			Basic	Intermediate	Advanced
Design of robot's behavior	Generic, Scientific, Continuum	Design plan of RoboDog's behaviors	Incorrect design of behaviors	Incomplete design of behaviors	Complete design of behaviors
Description of robot's behavior	Dynamic, Generic, Closed-loop	Description of RoboDog's reactive behaviors by "if-then" diagrams	Use of anthropomorphic language	Use of "mixed language"	Use of technological language

4 Findings

In the study we analyzed worksheets performed by 89 students participated in the workshop. In the 1st assignment of the 1st worksheet only 15 % of the students identified the RoboCroc's simple and complex behaviors, while 45 % were at the basic level (Table 5). The analyses of the 1st assignment of the 2nd worksheet, comparative to the similar assignment in the 1st one, indicated a significant increase in the students' performances of this task: 60 % of the students not only identified the robot behaviors but also correctly design them and only 25 % remained at the basic level, while in the 1st worksheet the correspondent percentage was 15 and 45 %.

Our findings from the analysis of "if-then" statements on the cause-effect relations between robot's response and the stimulus compiled by the students in the 2nd assignment were as follows (Table 6). In the RoboCroc worksheet most of the students described these statements using the anthropomorphic language (in terms of body parts), while only 7 % compiled at the advanced level using the technological language with more than half of the students (54 %) at the basic level. In the RoboDog worksheet, already 29 % achieved the advanced level with only about one third (35 %) remained at the basic level.

Table 7 presents and juxtaposes the evaluations of performance of the 3rd assignment in the worksheets related to RoboCroc and RoboDog. As shown, for the RoboCroc worksheet, the first one done by the students, only 15 % performed the task of identification robot components at the advanced level, while 54 % were at

Table 5 Inquiry of robot behaviors

Performance levels	Percentage of students	
	1st worksheet	2nd worksheet
Identifying simple behaviors	45	25
Incorrect identification of simple and complex behaviors	40	15
Complete identification of simple and complex behaviors	15	60

Table 6 Description of robot behavior

Performance levels	Percentage of students	
	1st worksheet	2nd worksheet
Use of anthropomorphic language	54	35
Use of mixed language (anthropomorphic and technological)	40	36
Technological language	6	29

Table 7 Identification of robot components

Performance levels	Percentage of students	
	1st worksheet	2nd worksheet
Noting in terms of body parts	54	22
Incomplete specification of components	31	49
Complete specification of components	15	29

the basic level. In the RoboDog worksheet, offered at the later stage of the workshop, already 29 % of the students demonstrated the advanced level and only 22 % remained at the basic level.

Figure 4 presents students' visual expressions and notes made by the second graders at the workshop and collected by the school teachers. The drawings show students' engagement and the notes indicate that they enjoyed the learning activities (Fig. 4).

Fig. 4 Students' expressions of the workshop

5 Conclusions

Our study indicates that the proposed approach, based on learning through interactions with animal-like robots, can be effectively implemented in the museum environment. The approach engaged children in the early years of elementary school in joyful learning in which they gained understanding of the robot system and the principles of its programming and operation. To answer the research question, we applied the Richmond's model to evaluate students' progress in the development of the systems thinking skills. We found that activities in inquiry and design of robot behaviors can foster students' systems thinking skills. The progress was achieved in all the categories of systems thinking, defined by Richmond. Children's reflections on the workshop activities were very positive. They were closely engaged in and excited by the interactions with robots and gladly collaborated in inquiry and creation of robot behaviors. Based on our positive experience in the museum environment, we recommend to examine the proposed approach also in formal education settings.

Acknowledgments The study was supported by the Technion Center for Robotics and Digital Technology Education and by the MadaTech Gelfand Center for Model Building, Robotics and Communication.

References

1. Hipkins, R., Bull, A., Joyce, C.: The interplay of context and concepts in primary school children's systems thinking. In: Hammann, M., Reiss, M., Boulter, C., Tunnicliffe, S.D. (eds.) Biology in Context. Institute of Education, University of London, Learning and teaching for the twenty-first century (2008)
2. National Standards: STEM Standards, (March 2015) http://www.clexchange.org/curriculum/standards/stem.asp
3. Rusk, N., Resnick, M., Berg, R., Pezalla-Granlund, M.: New pathways into robotics: Strategies for broadening participation. J. Sci. Educ. Technol. **17**(1), 59–69 (2008)
4. Verner, I., Polishuk, A., Klein, Y., Cuperman, D., Mir, R., Wertheim, I.: Robotics education through a learning excellence program in a science museum. Int. J. Eng. Educ. **28**(3), 523–533 (2012)
5. Shibata, T., Wada, K., Tanie, K.: Subjective evaluation of seal robot at the national museum of science and technology in Stockholm. In: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, Millbrae, California, pp. 397–402 (2003)
6. Hashimoto, T., Kobayashi, H., Polishuk, A., Verner, I.: Elementary science lesson delivered by robot. HRI 133–134 (2013)
7. Horn, M.S., Solovey, E.T., Jacob, R.J.K.: Tangible programming for informal science learning: Making TUIs work for Museums. Proceedings of 7th International Conference on Interaction Design and Children IDC, pp. 194–201. ACM, NY (2008)
8. Nourbakhsh, I., Hamner, E., Ayoob, E., Porter, E., Dunlavey, B., Bernstein, D., Crowley, K., Lotter, M., Shelly, S., Hsiuand, T., Clancey, D.: The personal exploration rover: educational assessment of a robotic exhibit for informal learning venues. Int. J. Eng. Educ. **22**(4), 777–791 (2006)

9. Riess, W., Mischo, C.: Promoting systems thinking through biology lessons. Int. J. Sci. Educ. **32**(6), 705–725 (2010)
10. Richmond, B.: Systems thinking: critical thinking skills for the 1990s and beyond. Syst. Dyn. Rev. **9**(2), 113–133 (1993)
11. Thomas, D.R.: A general inductive approach for analyzing qualitative evaluation data. Am. J. Eval. **27**, 237–246 (2006)

Robot Moves as Tangible Feedback in a Mathematical Game at Primary School

Sonia Mandin, Marina De Simone and Sophie Soury-Lavergne

Abstract We study how elementary school pupils give sense to the moves of a mobile robot in a mathematical game. The game consists in choosing 3 numbers out of 6, whose sum is a given target number. The robot moves on a game board have been implemented to provide pupils with a tangible feedback about their answer. We have studied strategies of pupils to solve the problem and their evolution. Our methodology included interviews, aloud verbalization and video observations of 28 pupils in grade 1 and 2 while they are playing. The pursuit of a mastery goal encourages a trial and error strategy for only some of the pupils. We conclude that some aspects of the moves of the robot, like its position, are perceived as a form of help and not as a threat, even if they are only partially understood.

Keywords Robot · Tangible feedback · Learning · Solving strategy · Mathematics

1 Introduction

The OCINAEE project[1]—Connected Objects and Digital Interfaces for Learning at Primary School—aims to explore and design mathematics learning situations with a system of connected objects. The system is a set of interacting devices either concrete like cards or a game board, or digital, like a smartphone. Connection

---

[1] Funded by the French Bank for Public Investments, it is a partnership between two companies, digiSchool and Awabot, and two public institutions, Erasme and the French Institute of Education.

S. Mandin (✉) · M. De Simone · S. Soury-Lavergne
Eductice/S2HEP, IFE, ENS-Lyon, Lyon, France
e-mail: Sonia.Mandin@ens-lyon.fr

M. De Simone
e-mail: Marina.De-Simone@ens-lyon.fr

S. Soury-Lavergne
e-mail: Sophie.Soury-Lavergne@ens-lyon.fr

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_22

between the two classes of objects is implemented by a mobile robot that can read some tangible objects such as playing cards or any specific printed material. In this project, we develop several games, which address mathematics for pupils from grade 1 to 6.

Our research deals with the interface of such a complex system of tangible and digital objects including a robot: how are users taking action on the system? How does the system provide feedback to the user? What should the designers choose? How do users understand the feedback? We study specifically actions and feedback of OCINAEE system in the tangible world, which are different from actions and feedback on a computer screen. Tangible objects like a set of cards and a robot moving on a board may be means of action and feedback.

Our theoretical framework crosses cognitive psychology and mathematical education, while mainly referring to objects that belong to computer sciences. After a presentation of this theoretical framework, which has framed the design of our games and the experiment we present here, we will describe the OCINAEE system for one of the games and the study of how the implemented feedback is understood by 28 French pupils in grade 1 and grade 2.

2 Theoretical Framework

2.1 Handling and Tangible Devices

The physical handling of concepts promotes the children's learning including the transfer of learning [1]. Many research works in education, psychology or specific to mathematics education have shown that mathematical concepts are bodily embedded and concepts are developed on tangible and physical manipulations [2, 3]. Concreteness is also used as a way to produce tangible feedback in the theory of didactical situations in mathematics [4].

Also, new interfaces have to be figured out as "*cognitive tools for promoting thinking and adaptive learning, rather than only emphasizing technology for interpersonal communication*" [5] (p. 25). But tangible and connected objects allow to link physical and virtual worlds. Therefore, in a learning perspective, we want to study their possible uses and the actions, manipulations and feedback they create.

Tangible interfaces can be described as objects handled by users and used to control a computer [6]. Using such interfaces in a learning situation, like a pedagogical game supports learning in different ways. Involvement of the pupils in the learning process is improved. Africano et al. [7] and later Kubicki et al. [8] have shown that tangibility through interactive tables allows to increase the active handling time of the concepts and a simultaneous activity and collaboration among pupils. Nevertheless, involvement is necessary but not enough to provoke learning. Tangible feedback provides information about mathematical solving strategy of the pupils. According to Brousseau's theory of didactical situations [4], learning situations are modeled by a game in which the possibility of different choices of actions for the player and appropriate material feedback are conditions for learning to occur. Moreover, pupils have to know by themselves if they have succeeded or not in solving the problem, independently from an external evaluation. Actions and feedback of the tangible world offer additional system of representations for mathematical concepts. According to Vergnaud [9] then Balacheff [10], mathematical concepts and knowledge are defined by a set of elements among which a semiotic system of linguistics and symbolic representations. Balacheff *"emphasis the importance of the manipulation of representation systems"* [10] (p. 8). Learning develops when using and translating from one system to another, including tangible experience. Sylla et al. [11] use cards with some chosen representations of concepts in their *Tangible interface for storytelling*. They are images to combine in order to create stories. Tangible cards support pupils in developing meaningful stories by easily changing their sequence.

Tangible objects and concrete phenomena may be used to drive the virtual system. Feedback concerns the reaction of the system toward the user. It is usually virtual, on the screen of a computer. Now with a mobile robot, the virtual system may drive the behavior of a concrete object and therefore produce tangible feedback. In the OCINAEE system, we have implemented a tangible feedback, which consists in the moves and the positions of a robot on a game board. It belongs to the concrete world and may translate some mathematical properties into a physical behavior. We want to study how learners interpret and give sense, a mathematical sense if any, to the tangible feedback of the robot.

2.2 Feedback

As anticipated above, we are interested in analyzing the feedback of the system on pupils. In general, we will refer to the term *feedback* as a return from action. We will introduce studies concerning feedback in different research fields. In particular, on the one hand, we consider feedback as "metacognitive" feedback and, on the other hand, as "didactical" feedback more focused on the mathematical task.

In research about games in general, integrated feedback is studied by Salen and Zimmerman [12]. For a meaningful game, users have to perceive feedback and consequences of their choices and actions on the game. Moreover, to improve players' experiences, Sweetser and Wyeth [13] highlight the importance of immediate feedback about the progression toward the goal of the game, for instance by making the score always accessible.

From a cognitive point of view and in a learning context, feedback aims to allow the learners to focus on the gap between their performance and the goals to reach [14, 15] by providing information on the accuracy of the answer or clues to reach the goals [16].

Noury et al. [17] link feedback and help, considering different types of feedback as helps. They highlight the relations between self-fulfillment, metacognitive judgment and nature of the helps used by learners. They distinguish between two types of helps: the instrumental helps providing clues to learners and the executive helps providing the answers. In their experiment, undergraduate students have used both types of helps after an error feedback, especially when they were following a mastery goal. However, learners use less instrumental help when they perceive it as a threat to their need for autonomy. They also avoid the executive help when they perceive it as a threat to their skills revealing their incompetence (performance-avoidance goals). There is no observed effects of the students' desire to demonstrate their skills (performance-approach goal) on the use of the helps. With OCINAEE, the moves of the robot can help pupils by showing the gap between their answer and the expected one. Thus the robot moves could be qualified as instrumental help. However, if this help is a necessary step to move on with the game and if it is not available on demand as in the experiment of Noury et al. [17] it can be perceived as a threat.

Rodet [18] distinguishes between cognitive feedback whose aim is the assessment of a work (taking into account the final result or the learner's approach), metacognitive feedback on cognitive processes of learners (in order to encourage to a personal reflection) and methodology feedback concerning appropriate strategies in order to improve later uses. These types of feedback are related to the actions, the cognitive processes and the knowledge of learners. Two types of feedback in Rodet's approach [18] are in line with a more didactical point of view. Didactical feedback is a response of the system that has some meaning from the learners' point of view in relation to the knowledge at stake [4]. Mackrell, Maschietto and Soury-Lavergne [19] distinguish between evaluation feedback in response to the achievement of the task and its success or failure and strategy feedback in response to a strategy of resolution in order to support evolutions of the strategy, therefore learning. Moreover, they highlight direct manipulation feedback as any immediate environment answer to the action of users. The two former types of feedback are built starting from the last one. From a didactical point of view, the design of feedback does not rely on knowledge about the learner as an individual, but rather on the components of the tasks and the knowledge involved in the solving strategies.

3 Description of the Game

We describe the implementation of the target number game with the OCINAEE system of devices, first by presenting the tangibles and virtual objects, then the scenario of the game, finally the design of feedback.

The target number is a problem solving situation which consists in choosing three numbers out of six, whose sum is a given target number. The target number is automatically chosen by the system, displayed on a smartphone. The six possible numbers are printed on six playing cards. The player has to choose three cards

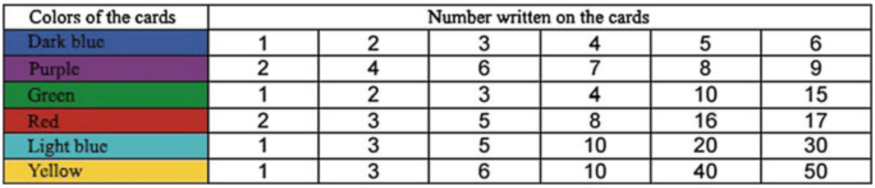
Fig. 1 The target number game devices and an example of a 6 card set (the *purple* one). To the *right*, submission of a card to the robot

among the six and to submit them to the system. A game is played in 6 untimed rounds, by a single player or a group.

The kit includes the following devices (see Fig. 1): a robot with a smartphone, a game board with a picture on which the robot moves, and 39 playing cards.

Technically, two sensors placed under the robot allow to scan any code on the game board and on the cards (the code is not visible by eye). At any time, the system knows the robot location and can determine its trajectory and positions. The robot eyes light up in different colors. The cards are grouped in 6 sets of 6 cards. Each card presents a number and each set of cards is associated with a color. When users select a card and want to submit it to the system, they scan it under the robot (see Fig. 1, right). Three other cards are available to "validate" the scanned cards, to "cancel" the scan of all the cards and to "listen" to the target number. The picture of the game board represents a landscape with five characters aligned on the skyline. They materialized different final positions of the robot.

At the beginning of the game, once pupils have placed the robot on the game board, the system displays a target number on the smartphone with a colored background. This indicates the color of the set of cards to use (see Fig. 2). Pupils have to select and scan three cards whose sum has to be equal to the target number. The robot eyes light up each time a card is scanned: in white for a correct scan and in red in case of a card of the wrong color or twice the same card. There is no immediate feedback if the number of cards is not three.



Colors of the cards	Number written on the cards					
Dark blue	1	2	3	4	5	6
Purple	2	4	6	7	8	9
Green	1	2	3	4	10	15
Red	2	3	5	8	16	17
Light blue	1	3	5	10	20	30
Yellow	1	3	6	10	40	50

Fig. 2 Numbers on each of the six sets of cards

Then, pupils have to submit the “validate” card to get an evaluation from the system. Hence, the system computes the sum of the numbers on the chosen cards. According to this sum, the robot moves to one of the different characters on the game board. The robot final position is: (i) on the marmot when the sum is lower and far from the target number, (ii) on the sheep when it is lower but close to the target number, (iii) on the shed when it is equal to the target number (success), (iv) on the snowman when it is above and close to the target and (v) on the yeti when the sum is above and far from the target. The characters are aligned to mediate the number line.

Additional feedback is implemented. The smartphone displays a congratulation message or messages telling the pupils that their sum is too small or too big or that they have scanned a wrong number of cards (different from 3). In case of errors, pupils have two other attempts to try again. The robot finally performs a “dance” when pupils have succeeded in all of the six rounds of the game.

4 Research Questions

We are going to study tangible feedback through its effect on pupils’ solving strategies and through the way pupils transform it into a possible help. The tangible feedback in OCINAEE system is constituted by a robot, which moves and reaches a final position on the game board.

Our question concerns the perception and exploitation of this tangible feedback by pupils. It has been designed as an evaluation feedback produced when requested. Does the position of the robot inform pupils about the validity of their answer and does this feedback have consequences on pupils’ solving strategies? In other words, we wonder if a tangible feedback may be an evaluation and/or a strategy feedback.

Moreover, pupils can consider feedback as possible helps. But according to the kinds of goal they are following, mastery and/or self-fulfillment, they may consider instrumental help as a threat to autonomy, therefore ignoring the feedback. Consequently, in our observations, we are going to identify pupils’ goal and their perception of threat through the feedback.

5 Methodology: Participants and Experimental Setting

To answer the research questions above, we have conducted an exploratory study with some of the pupils of the 35 teachers involved in the OCINAEE project. The OCINAEE project concerns 39 classes from grade 1 to 6 in 14 schools and 4 junior high schools. Ongoing studies concerning all the classes will also drive us in the improvement of the games and the robot moves as feedback. As we are interested in mathematical skills of the curriculum of the early classes, we limit our study to pupils in grades 1 and 2. The 28 pupils of the experiment come from 5 different classes, 8 pupils belong to 2 grade 1 classes and 20 pupils belong to 3 grade 2 classes. The observations were held during three weeks around Christmas holidays. To minimize the effect of isolating one pupil at a time, we have observed pairs of pupils, which indeed is the usual configuration chosen by teachers when they use the game in their class. In each class, pupils have been randomly selected two by two and extracted from the classroom for about one hour corresponding to the duration of the different steps of the experiment. These pupils have already played with other versions of OCINAEE games, under the supervision of their teachers. They have also played the target number game, but without any moves of the robot. These teaching sequences could not be controlled. However it has no consequences on our experiment, because the pupils use the dynamic version of the game for the first time.

The experimental setting consists in four steps: think-aloud training, interview before playing framed by a short questionnaire, play, interview after playing framed by a long questionnaire. The questionnaires have a total of 19 questions (6 open-ended and 13 close-ended questions with requests for explanations sometimes). Some questions aim only to open dialogue or allow pupils to recall the play (e.g., *Have you played to target number before? Did you see the robot move?)*. Other questions propose some choices to help the children to answer (e.g., *When you were playing, you would: (1) like to show that you are doing well? (2) avoid showing you can make errors (3) improve yourself in calculations?*)

Think aloud was a mean to perceive the pupils' strategies during the play. It has been used in many areas, like to identify processes involved in the understanding of a text [20], the writing process [21] or mathematical problems solving [22]. It is not an easy task for young pupils. To make them more comfortable, we trained them with 2 exercises that we assumed to be familiar. The first is an individual exercise of continuation of a necklace in accordance with a color code. The second is a game of noughts and crosses. It is a strategic exercise, which requires to alternate the speaking times. Throughout training, when pupils were not verbalizing aloud their thoughts, the researcher was fostering it with questions (e.g., *what are you doing? Where do you put your cross? Why?*). The training lasted approximately 15 min.

The first interview aims to catch how pupils understand the game board and the dynamic version of the game when they discover it for the first time. In particular, we have asked the pupils to comment on the usefulness of the characters, including the shed, on the game board. Just after the first interview and before the play, a short explanation of the breakpoints has been provided to the pupils by the researcher.

At the beginning of the game, pupils have selected the range of the target numbers (number target up to 20, 40 or 100). However the difficulty of the game doesn't lie in the size of the target number but in some features of the solution, like the possibility or not to find two cards whose sum is a multiple ten or the presence of distractive cards. During the play, we have videotaped the pupils. In the analysis, we focus on the handling of the playing cards and their thinking aloud. Despite the think aloud training, we have anticipated two risks. A first risk is the simultaneous recording of two pupils' voices and therefore difficulty to understand their talk afterwards. A second risk is that despite training, thinking aloud remains difficult for young pupils. We therefore carried out a combined analysis focusing on both their speech and their actions. All data have been manually analyzed according to our theoretical framework.

The after game interview is more complete. It allows to compare the pupils' declaration to their actual achievement during play. We analyze the self-fulfillment of the pupils (mastery vs performance goals) and their representation and exploitation of the robot moves on the game board. The after game interview was administered to 25 of the 28 pupils observed (it was not possible to run the interview with 3 pupils).

6 Results

The results focus on the interpretation of the game board by the pupils, their involvement during the game and the evolution of their solving strategies.

6.1 Interpretation of the Game Board and the Moves of the Robot

The answers of the pupils to the first interview show that the positions represented by the different characters on the game board have a meaning within the task only for grade 2 pupils.

In grade 1, 2 out of 8 pupils separate the game board in two opposite team sides and identify the different spots as symbolizing a failure or a success for each team. The other pupils propose purely fictional explanations. 8 out of 20 grade 2 pupils interpret these characters as positions of the robot, meaning that the sum of the numbers is too big or too small with respect to the target number. Among them, 4 pupils perceive a shorter and more rarely a longer distance to target number when the robot stops on the Yeti, and 3 pupils correctly interpret the positions represented by the characters. One grade 2 pupil reacts as grade 1 pupils above-mentioned, the others do not give any explanation.

The most frequent explanation of the aligned positions of the characters is the facilitation of the robot move (6 pupils out of 28). This justification highlights the focus of the pupils on the robot. But 4 pupils in grade 2 mention that characters are sorted in ascending order according to the sum of the chosen numbers. This can be analyzed as an initial mathematical interpretation going toward the concept of number line. The others were undecided or gave off topic answers.

After the play, pupils still have misconceptions of the game board, with a better interpretation in grade 2. The absence of materialization of the starting area led to a confusion with the first position of the robot represented by the marmot (6 pupils out of 25). The shed is the best understood character (19 pupils out of 25) compared to the other four spots (correctly interpreted by 13 to 16 pupils out of 25).

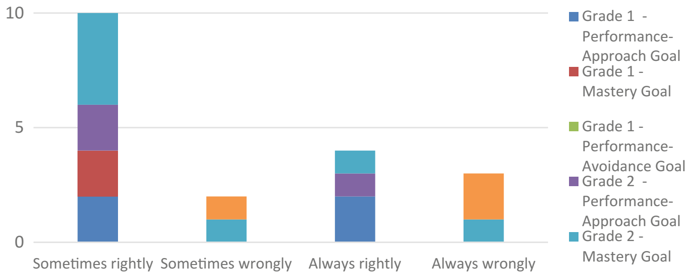


Fig. 3 Distribution of pupils according to their answer to the question of achievement (*have you found the answer at the first attempt always/sometimes or never?*) in comparison with their achievement observed in the game (*rightly* if the answer corresponds to the observation else *wrongly*) and their goals of self-fulfillment

6.2 Involvement and Goals of Pupils During the Play

As shown in Fig. 3, interviews after playing reveal that 4 out of 6 grade 1 pupils try to progress (approach performance goal) while 13 out of 19 grade 2 pupils try to improve their skills in computing (mastery goal) and only 3 grade 2 pupils pursue a performance-avoidance goal. Their main goal is avoiding errors. This may explain why they wrongly answer the question about their success at first attempt (one of them wrongly read the target number and found a correct sum according to the targeted number). Indeed, none of the grade 1 pupils were pursuing performance-avoidance goal and they all answered correctly the same question.

6.3 "Trial and Error" or "Compute and Check" Strategies

Pupils explaining what they do in order to play mainly refer to their mental processes: count, search, think (4 grade 1 and 16 grade 2 pupils). Scanning the cards is also frequently mentioned (2 grade 1 and 5 grade 2 pupils). They rarely mentioned the testing of a combination of cards (1 pupil only at each grade). These declarations are consistent with the observed behaviors during the play. Videos allowed us to count the number of pupils showing a trial and error strategy. These pupils do not anticipate the sum of their combination of cards but first submit the cards to the robot to get its feedback. We call them "testers". We oppose this strategy to a compute and check strategy. In this case, pupils add the numbers on the cards before scanning them, they are "checkers". There are more "testers" than we presupposed according to the pupils' answers in the interview (7 pupils out of 25) but they are much fewer than the "checkers" (16 pupils out of 25). Moreover, the "trial and error' strategy has been often triggered by the experimenter to help some pupils facing difficulties and hesitations. The "testers" also seem to be mostly pupils pursuing a mastery goal (5 "testers" out of 7) while the "checkers" are better

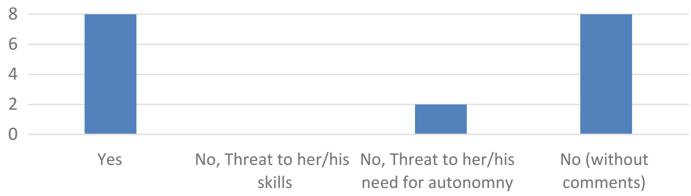


Fig. 4 Distribution of pupils according to the perception of the moves of robot

distributed between the different goals of self-fulfillment (9 “checkers” pursue a mastery goal out of 16).

6.4 *Effects of the Robot Moves as Feedback on the Pupils’ Strategies*

In case of successive trials, it is expected that the robot moves have consequences on the evolution of strategies like the new selection of playing cards.

But, according to the interview after play, less than half of pupils speaks of the robot moves (8 out of 19 pupils, see Fig. 4). Five of them mention the moves for checking if their sum of numbers is too small or too large or right or wrong. The other 3 pupils do not bring precision. Indeed, as we see below, pupils modify their combinations of cards according to the overshoot of the target number rather than to the distance between their sum and the target number. Among the 11 pupils who do not claim to have used the robot, 7 pupils do not understand the meaning and therefore the usefulness of the robot positions and moves. Only two pupils give a reason for not using them while they correctly understand them. Both pupils claim they want to succeed by themselves. Thus, the robot move is not generally perceived as a threat neither on need for autonomy nor on skills.

The effect of robot feedback on strategies evolution is not direct. Among the 21 pupils who had another trial after errors (not due to a bug, a wrong number of playing cards or scanning the same card twice), 10 pupils affirm they start again from their previous combination by changing one or two cards sometimes or often (3 grade 1 pupils and 7 grade 2). For them, the robot moves work as a strategy feedback. The other pupils start again as with a new target number. However, in the 25 s or third attempts concerning 10 pairs of pupils, 18 combinations involve the change of only one card. In 11 of these trials, the new card replaces the smallest card of the previous combination (see Fig. 5). The number of cards that are modified between two attempts seems very strongly linked to the sum, smaller or bigger than the target number. If the sum exceeds the target number, pupils usually change two playing cards (3 cases out of 7). With a sum smaller that the target number, pupils change only one playing card (14 out of 18), usually the smallest (10 out of 14).

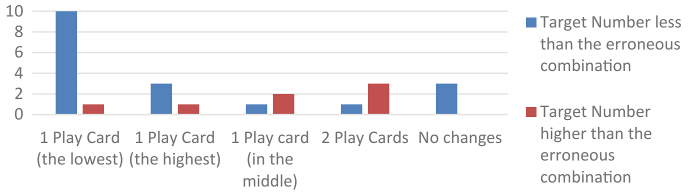


Fig. 5 Distribution of changes of cards according to the value of the previous combination

These observations show that these pupils seem guided by a strategy that leads them to exceed the target number and then get closer by changing the smallest card as if it would "go down" more gradually. The game as a whole allows to highlight the strategies and foster their evolution. But, for the moment there is no evidence that strategies evolve towards more efficient ones.

7 Conclusion

In this paper, we have described an experiment which aimed to highlight the interpretation and exploitation of a robot moves on a game board as tangible feedback by grade 1 and grade 2 pupils. The target number game is played with cards that has to be scanned under a robot. According to the sum of the numbers printed on the cards, the robot moves to a position, indicating the distance between the sum and the target.

We have noted several differences between the pupils according to their grade. First, the game board is designed so as to mediate the number line. However the meaning of the picture and the aligned position of the characters are not understood by all pupils, especially the pupils in grade 1. They do not make connections between the game board and the mathematical concepts involved in the game. Second, it concerns the goals of self-fulfillment. The grade 1 pupils want to demonstrate their skills (performance-approach goal) while most of the grade 2 pupils want to improve them (mastery goal) and only some of them want to avoid showing they may make errors (performance-avoidance goal). These different goals of self-fulfillment appear to influence the pupil strategies in the way they use the robot. Some of them test their combinations of cards without anticipation whereas others scan their cards only once they are sure of their sum. Most of the "testers" pursue a mastery goal as if they were aware that the robot may supports the evolution of their strategies. Nevertheless, it remains interesting that the help provided by the robot moves is not perceived as a threat. Therefore, we can assume that most of the pupils have not ignored the tangible feedback of the system. Finally, tangible feedback is an evaluation feedback for about three quarter of the pupils. The others rely on the message on the smartphone instead of the robot position to evaluate their answer. Also tangible feedback is a strategy feedback for only about half of the pupils, the ones who modify their strategy according to the robot moves. This is not surprising because many of them have not understood the meaning of the robot positions except for the shed. Didactically, it is also interesting to observe that pupils modify their invalid combination according to how it exceeds the target number and not according to its distance to the target number. If the sum exceeds the target number, only the lowest number is changed, otherwise two cards are most often replaced. This strategy isn't the most efficient one and asks for further adaptations of the game and its feedback.

Actually, the evolution of pupils' strategies could also be obtained by a personalization of the learning situation, based on a learner profile [23]. In our game, for pupils using a strategy based on considering if the sum is just exceeding the target number instead of considering the distance between the sum and the target number, the system should provide new target numbers that make pupils aware of the limitation of their previous strategy.

This study is interesting to frame research about how concrete objects and phenomena produce tangible and immediate feedback and it has to be continued in order to study long-term evolution of pupils' strategies.

References

1. Martin, T., Schwartz, D.L.: Physically distributed learning: adapting and reinterpreting physical environments in the development of fraction concepts. Cogn. Sci. **29**(4), 587–625 (2005)
2. Lakoff, G., Nunez, R.: Where mathematics comes from: how the embodied mind brings mathematics into being. Basic Books, New York (2000)
3. Edwards, L., Radford, L., Arzarello, F.: Gestures and multimodality in the construction of mathematical meaning. Educ. Stud. Math. **70**(2) (2009)
4. Brousseau, G.: Theory of didactical situations in mathematics. Springer, Netherlands (1997)
5. Oviatt, S.: Designing digital tools for thinking, adaptative learning and cognitive evolution. CHI, Vancouver, Canada (2011)
6. Mellet-d'Huart, D., Michel, G.: Réalité virtuelle et apprentissage. In: Grandbastien, M., Labat, J.-M. (eds.) Les environnements informatiques pour l'apprentissage humain. Traité IC2 Information Commande Communication. Hermes (2006)
7. Africano, D., Berg, S., Lindbergh, K., Lundholm, P., Nilbrink, F., Persson, A.: Designing Tangible Interface for Children's Collaboration. CHI, Vienna, Austria (2004)
8. Kubucki, S., Pasco, D., Arnaud, I.: Using a serious game with a tangible tabletop interface to promote student engagement in a first grade classroom: a comparative evaluation study. Int. J. Comput. Inf. Technol. **4**(2), 381–389 (2015)
9. Vergnaud, G.: The theory of conceptual fields. Hum. Dev. **52**, 83–94 (2009)
10. Balacheff, N.: cK¢, a model to reason on learners' conceptions. In: Martinez M. V., Castro Superfine A. (eds.) PME-NA Psychology of Mathematics Education, North America Chapter. pp. 2–15. Chicago, IL, USA (2013)
11. Sylla, C., Branco, P., Coutinho, C., Coquet, E., Skaroupka, D.: TOK—a tangible interface for story's telling. CHI 2011, Vancouver, Canada (2011)
12. Salen, K., Zimmerman, E.: Rules of Play: Game Design Fundamentals. The MIT Press, Cambridge, MA (2004)

13. Sweetser, P., Wyeth, P.: GameFlow: a model for evaluating player enjoyment in games. Comput. Entertain. **3**(3), 1–24 (2005)
14. Kluger, A.N., DeNisi, A.: The effects of feedback interventions on performance: a historical review, a meta-analysis, and a preliminary feedback intervention theory. Psychol. Bull. **119**(2), 254–284 (1996)
15. Hattie, J., Timperley, H.: The power of feedback. Rev. Educ. Res. **77**(1), 80–112 (2007)
16. Shute, V.J.: Focus on formative feedback. Rev. Educ. Res. **78**(1), 153–189 (2000)
17. Noury, F., Huet, N., Escribe, C., Sakdavong, J.-C., Catteau, O.: Buts d'accomplissement de soi et jugement métacognitif des aides en EIAH. Environnement Informatique pour l'Apprentissage Humain (EIAH 2007), pp. 293–298. INRP, France (2007)
18. Rodet, J.: La rétroaction, support d'apprentissage? Revue du Conseil Québécois de la Formation à Distance. **4**(2), 45–46 (2000)
19. Mackrell, K., Maschietto, M., Soury-Lavergne, S.: The interaction between task design and technology design in creating tasks with Cabri Elem. In: Margolinas, C. (ed.) ICMI Study 22 Task Design in Mathematics Education, pp. 81–90. Oxford, UK (2013)
20. Magliano, J.P., Millis, K.K.: Assessing reading skill with a think-aloud procedure and latent semantic analysis. Cogn. Inst. **21**(3), 251–283 (2003)
21. Hayes, J.R., Flower, L.S.: Identifying the organization of writing processes. In: Gregg, L.W., Steinberg, E.R. (eds.) Cognitive Processes in Writing, pp. 3–30. Erlbaum, Hillsdale (1980)
22. Rosenzweig, C., Krawec, J., Montague, M.: Metacognitive strategy use of eight-grade students with and without learning disabilities during mathematical problem solving: a think-aloud analysis. J. Learn. Disabil. **44**(6), 508–520 (2011)
23. Mandin, S., Guin, N., Lefevre, M.: Modèle de personnalisation de l'apprentissage pour un EIAH fondé sur un référentiel de compétences. EIAH'15, Agadir, Maroc (2015)

Personalizing Educational Game Play with a Robot Partner

Mirjam de Haas, Iris Smeekens, Eunice Njeri, Pim Haselager, Jan Buitelaar, Tino Lourens, Wouter Staal, Jeffrey Glennon and Emilia Barakova

Abstract Personalization of educational and behavioral training to the developmental stage of the individual child is common practice in educational and therapeutic settings. Research on robot-based education training is only just starting to adopt this approach. We present a pilot study on a behavioral intervention design in which Pivotal Response Training (PRT) elements are embedded into a game played by a robot and a child. Seven game levels are designed to cover different levels of communication skills that are targeted by PRT. The levels do not differ with respect to the logical steps in the game that the children should take, only with respect to the social competence that the child has. The behaviors displayed at each stage were observed and analyzed using qualitative and quantitative methods. Our results indicate that the more socially challenging a game is, the happier children are and the more children engage with playing the game, although the game challenge remains the same.

M. de Haas (✉)
Tilburg Center for Cognition and Communication,
University of Tilburg, Tilburg, The Netherlands
e-mail: mirjam.dehaas@uvt.nl

M. de Haas · E. Njeri · E. Barakova
Department of Industrial Design, Eindhoven University of Technology,
Eindhoven, The Netherlands
e-mail: E.N.Mwangi@tue.nl

E. Barakova
e-mail: e.i.barakova@tue.nl

M. de Haas · P. Haselager
Department of Artificial Intelligence, Radboud University Nijmegen,
Nijmegen, The Netherlands
e-mail: w.haselager@donders.ru.nl

I. Smeekens · P. Haselager · J. Buitelaar · J. Glennon
Donders Institute for Brain, Cognition and Behaviour,
Radboud University Nijmegen, Nijmegen, The Netherlands
e-mail: iris.smeekens@radboudumc.nl

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent
Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_23

Keywords Social robotics • Initiation of interaction with a robot • Games with robots • Personalizing educational games • Social competences for children with ASD

1 Introduction

Increasingly more attention has been given to the use of social robots in education and behavioral training for typically developed children and for individuals with Autism Spectrum Disorders (ASD) [1–4]. The use of robots for ASD education and care is one of the most promising use of robots, yet a tremendous amount of theoretical and practical work is required to make it come true.

Previous studies show that children with ASD normally respond well to robots, sometimes even prefer them to humans in similar settings [2]. This has been attributed to the fact that social robots are easier to interact with; due to their simpler facial and gestural features that do not overwhelm the child. In addition, it was shown that the effectiveness of question asking training was equally high when performed with a robot or a human trainer [5].

The robots are programmed with social behaviors designed to elicit fundamental social skills such as eye contact, turn taking, and imitation among others. Children with ASD possess a number of social deficits including: difficulties in recognizing nonverbal communication cues such as body language, lack of attention, inability to switch from focusing on one thing to another, difficulty in communicating and reciprocating in conversations and a lack of social initiations [6]. Recent research reports enhanced levels of social behaviors such as attention and engagement, [1, 4, 5, 7] enhanced social initiations skills [1], imitation [2, 8], and improved turn taking behaviors. Although these shortcomings are typical for individuals with ASD, the typically developing children will also benefit from a personalized training corresponding to their level of language comprehension.

In therapeutic and educational settings, game sessions with robots are often used and have been proven to be an effective way of training social skills with children

---

J. Buitelaar
e-mail: j.buitelaar@psy.umcn.nl

J. Glennon
e-mail: j.glennon@cns.umcn.nl

I. Smeekens · J. Buitelaar · W. Staal
Karakter Child and Adolescent Psychiatry, University Centre Nijmegen,
Nijmegen, The Netherlands
e-mail: w.staal@karakter.com

T. Lourens
TiViPE, Helmond, The Netherlands
e-mail: tino@tivipe.com

[5, 9, 10]. In the existing educational and training practices there has been little attention for the personalization of the training to the developmental stage of the individual child. We have adopted the Pivotal Response Training (PRT) framework which differentiates the level of training of the child to the child's level of communication. This framework targets the children's pivotal areas of development instead of targeting individual behaviors. This results into improvements in other pivotal areas that are not directly targeted. To make the training scenarios more individualized to the child's development stage, 7 levels of game scenarios have been created that target different communication skills, varying in difficulty (competence levels). These scenarios help the children to increase their individual level of communication.

In the current pilot study we investigate the effects of robots on children who are typically developing, but do display differences in the level of proactive behaviors and speech development. With our pilot study one of the games designed to be tested with children with ASD, a card game, is analyzed. We aim to improve the overall design of the personalized robot behavior within the games, and specifically the flow of the interaction within each level of the game. The influence of the therapist who is present during the training sessions is also evaluated.

2 Experimental Design

The main aim of the current study is to find an effective way of integrating PRT principles within a game scenario, to facilitate a playful robot-child interaction at the communication level of the child. Diehl et al. [3] argued that an experimental setup, where the teacher or therapist is in the same room, should be given more attention. Moreover, Mehrabian [11] suggested that a teacher or therapist that is present at the robot session can help the child develop a more detailed conversation, or give the child feedback. However, these papers do not discuss the possible decrease of learning efficiency caused by the interference of the teacher or a therapist at the moments when the child and the robot are successfully engaged in interaction.

2.1 Study Setup

The robotic platform used for this study is the NAO humanoid robot, with 25 degrees of freedom, 58 cm in height. NAO has simplistic facial features, with only a mouth and eyes, and its face resembles to the age of a child's face. Some of the robot behaviors that were used in this game include text to speech functions, hand gestures, and NAO LEDs. We used the same experimental setting as described in [1]: the robot was placed in front of the child and the teacher or a therapist using a laptop was seated next to the robot. TiViPE, which is a graphical programming environment was used to program the robot and for interaction during the sessions

[12]. A special interface was used for real-time interaction between robot and child and was connected to the previously programmed interaction scenario. The pre-programmed scenario consisted of a dynamical system of behaviors for complex interactions, emotions and behaviors. This interface for a real-time interaction through speech and simple movements was created as a part of this program especially for this experiment.

During the experiment, the therapist was in the same room to control the robot's responses to the verbal cues of the child. The robot was used during a card game (Kwartet) to prompt the children in initiating actions (asking for a card), to react to questions and to give rewarding expressions afterwards.

2.2 Competence Levels

The children were divided in different social competence levels, based on how many times the children asked the robot questions, asked for help or made a statement and reacted on the cues of the robot during the introduction phase. The social competence level corresponded with a higher level of the game. At a higher level of the game the robot made the social interaction harder for the children. The robot would ask for more complex initiations (i.e. social instead of functional initiations). For example in the lowest level children only needed to ask for the cards, but in a higher level they also needed to protest when the robot intentionally performed a wrong move or tried to cheat.

Scenarios were developed for the different levels, incorporating three versions of a card game, which provided the child a choice for a game, and encouraged child initiative. These scenarios consisted of the flow of interaction between the robot and the child. Within the game scenarios, the therapist incorporated different levels of help (prompts) to the child that are provided by the robot. The therapist had the chance to type in extra robot speech utterances using a text-to-speech interface connected to the programmed robot scenario and these speech utterances could also be accompanied by corresponding gestures.

2.3 Participants

Eighteen children participated in this experiment (aged 5 to 8, mean 7, 15 boys and 3 girls). Table 1 displays the levels of the game and the number of the children per level. None of the children fitted in the level 1, 6 or 7. Two children were excluded from the analysis, one because of technical difficulties in the video material and the other child because (s)he did not understand the game and therefore required a lot of help from the therapist.

Table 1 The number of children assigned to a game level and description of the social competence levels

Level	No of children	Description of the social competency level
1	0	Speaking with 2/3 words
2	5	Asking for object/activity + multiple cues
3	8	Asking for object/activity + asking for help + protesting
4	3	Asking for help + protesting + www (what/when/where) question asking
5	2	www-question asking + protesting
6	0	www-question asking + asking through
7	0	Asking through + making statement + asking begin question

2.4 Observational Protocol

We designed a protocol for the observers in line with the objectives of this study. The protocol was divided into four sections. In the first section, the observer was required to note where the child looked (gaze direction) the observer could choose between the following options: (1) the child was looking at the robot, (2) looking at the experimenter or (3) shared their gaze towards a card with the robot. With these options, we aimed to examine the relationship between the gaze behavior and the level of pleasure or engagement during the game session.

In the second section, we asked the observer to rate the level of arousal of the child during the game session, based on three levels from neutral (0) aroused (+) to very aroused (++). The observer was required to look out for cues like clapping, winning gestures, raising hands or screaming or jumping among others.

In the third section the observer was asked to rate the valence observed during the interaction. Valence can either be negative or positive. Some of the cues that indicate negative valence include that the child closes eyes, looks down, yawns, expresses negative verbal responses, and is impatient. Positive valence cues are: smiling, positive verbal responses, winking teasingly at the robot etc.

In the last section the observer was required to rate the perceived emotions during the game, consisting of happy, angry, bored, confused, afraid, surprised or another emotion, inspired by the categorical scheme of Eckman [13]. To gather additional qualitative information and to verify the results we also asked for evaluation of the arousal level and the emotional valence, although we are aware of the consequences of mixing dimensional and categorical schemes for measuring emotions.

The observers were students at the university campus, who were taking a class on social robotics and as an initial exercise they were asked to rate videos. They were provided with the guidelines for observation. The observers viewed the videos on a computer and record their observations on the provided paper forms. The observers were allowed to stop the video to record their observations or to view it again if necessary. A total of 16 observers coded the videos and every child was observed by two observers each. The reliability of the observers can be found in the results section. Furthermore, we used the mean of all the scores to process the data.

2.5 Experimental Procedure

Every child played a card game with the robot. Introducing the robot to the children prior to the study was added as a first step to the experiment as robots are very unusual and exciting as playing buddies, but it also provided information to determine the social competence level of the children. The whole procedure took approximately 40 min for every child of which 20 min was the duration of the actual card game. A digital camera was used to record all sessions. This camera was placed in a way that the face and the action of the child could be observed at all times. During the whole session the robot was present and was leading the playful interactions.

The same therapist was present at each session to control and assist the robot, but also to offer help when children asked for help. After the child finished his/her turn, the therapist could choose the next behavior for the robot on a laptop. If the action of the child was unexpected, the therapist could prompt the child to interact by typing an additional robot utterance at the specially created interface. This interface was connected to the overall scenario but also permits interactions and modifications of robot behaviors to be made at every moment. The notebook on which this interface was displayed was on the same table as the card game and the robot.

All sessions started with an introduction in which the children had to build a tower with blocks. The blocks were provided by the robot. The performance of the children in the introduction scenario resulted into one of the 7 levels of the game scenarios. After the introduction the robot would explain the rules of the card game to the child and ask the child for help placing his cards in the robot's card holder. The children were given the same card holder as the robot to put their cards in, so they would not be distracted with holding the cards. The purpose of the game was to get four cards of the same category and therefore the robot and a child would take turns asking for a card. When the child asked the robot for a card, the robot would point at that card so the child could grab it, which can be considered as a natural reward. In the other case the robot would ask the child for a card and would also ask whether the child wanted to place the card in an empty place in his card holder.

2.6 Data Collection

All sessions were videotaped. From each video, several segments of 20 s interactions were extracted by the experimenter. These segments represent the interaction moments when a verbal utterance was produces either by the robot or by the child during the actual card game. There were two types of verbal interactions: the robot asked the child for a card and the child asked the robot for a card. These questions were accompanied with nonverbal behaviors that the robot used during the sessions. These behaviors can be described as cheering, pointing at card and turning its head to add more expressiveness to its speaking. This resulted in around 12 videos per child (depending on the course of the card game, some children had more interaction moments with the robot). The videos were watched in a randomized order by the observers. The coding of the videos was done in random order since we wanted the observers to remain uninfluenced by the context.

3 Results

The results section features first the qualitative analysis of the robot-child interaction. Second, the data analysis involved determining the inter-observer agreement (IOA). Further, we did a visual inspection of the data and calculated the mean of the results of the two observers for each competence level. Pearson correlation was used for determining whether the child was more confused when (s)he looked towards the therapist instead of the robot [14]. For the analysis, the mean ratings of the expressed emotions at the different levels of the children development were compared. A Shapiro-Wilk test was performed to test for normality of the data [15]. The data was not normally distributed, therefore a Kruskal-Wallis test was performed to check whether there was a difference over the levels [16]. For the levels for which a significantly difference was found a Bonferroni corrected one-tailed Mann-Whitney test was performed (significance border $p <= 0.0125$) [17].

3.1 Qualitative Analysis of the Robot Child Interaction

The behaviors that the robot used during the sessions can be described as speaking, cheering, pointing at card and turning his head to add more expressiveness to its speaking. Of the 16 observers 10 gave extra annotations about the robot's behavior.

Most observers noted that more variation of the interaction would improve the design. They repeatedly mentioned that the voice of the robot should match the robot's behavior, for example the voice did not match with the robot cheering. In addition, the robot had no variation/intonation in its voice, because of the imperfections in the computer generated voice. In one of the levels the robot would make a joke to provoke the children into protesting. The text-to speech engine is not able to enact laughter well instead it pronounces "hà-hà-hà-hà". These weaknesses were causing confusion by the children.

During the session the robot would ask for a card. If the children gave the right card the robot would confirm that fact, he would ask the child to place the card in an empty place in the card holder and finally, the robot would cheer. Another repeatedly noted observation was that the children already placed the card in the

Table 2 The reliability of recording calculated with the prevalence adjusted and bias-adjusted kappa. All scores higher than 0.4 are included in the analysis

Condition	PABAK	Condition	PABAK
Gaze to human	0.62	Angry	0.97
Gaze to robot	0.03	Bored	0.56
Joint attention	0.28	Afraid	0.81
Valence	0.42	Confused	0.51
Arousal	0.45	Surprised	0.80
Happy	0.40	Other emotion	0.36

card holder, and the robot would still say that the children had to place the card in the card holder. The cheer of the robot was at that moment redundant as the children waited for the robot's turn.

3.2 *Reliability of Recording*

The inter-observer agreement (IOA) was determined with the prevalence adjusted and bias-adjusted kappa (PABAK, see [18]). Table 2 shows the PABAK for every category (gaze, emotion dimensions and cognitive states). Agreement was defined when both observers identified the same behavior as present or rated the child equal on the scale. The agreement between observers was only moderate (Mean PABAK was 0.52). The observers showed a higher agreement on the categorical emotions than on the dimensional ratings (i.e. there was no agreement in the level of arousal, or valence). For the analysis we only considered the categories with a score higher than 0.4 (moderate agreement).

Furthermore, the mean of all scores was used to process the data. We did this under the impression that when the two observers disagreed on the score for a behavior the mean provides the shared view of the observers. For example when the child was slightly positive, some observers rated this child neutral and others positive. In that case the child had a score of 0 and 1, the mean score of these two observers (0.5) will be accurate enough to describe that the child was slightly positive. We still checked with a visual inspection of the data whether the observers did not disagree completely, since rating in both directions can result in neutral. In only 1 % of all the cases the observers disagreed completely.

3.3 *Levels of Social Competence*

It was expected that the dimensional measurements of emotion would be similar for the children in all different levels since the levels were designed for the social competence level of the children, while the presence of the different emotions can differ. Figure 1 shows in percentages the times observers rated the children in a certain emotion plotted versus the estimated level of social competence (which

Fig. 1 The percentage of the happy, bored, confused and surprised emotion per child developmental level. The observers were allowed to rate more than one emotion as the observed emotion in the child behavior

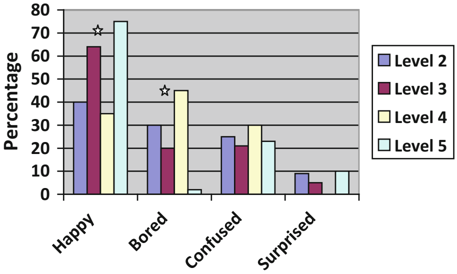
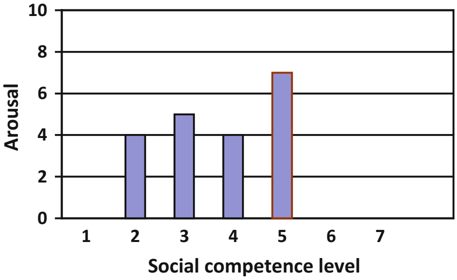


Fig. 2 The average of the perceived arousal in children's behavior at each competence level



corresponds to the level of the game). Figure 2 shows the average of the perceived arousal in children's behavior at each developmental level, which corresponds to the level of the game. The absolute arousal is highest in game level five, and lowest in game level four.

3.4 *Gaze Direction*

The observers were asked to record whether the child looked at the robot, the experimenter, or looked at the same card as the robot. It was expected that children, who were rated to look more at the experimenter, were also rated as more confused or more surprised and children who were rated to look less at the experimenter were

Table 3 The correlation and corresponding p-values between the observed gaze towards experimenter and observed confusion, surprised and happy. Confusion has a correlation of 0.358, which is not high, but still significant

	R	p
Gaze * confusion	0.358*	0
Gaze * surprised	0.037	0.635
Gaze * happy	−0.1	0.199

more often rated as happy in this game situation. As Table 3 shows, the confusion of the children correlated with the moments when children also looked at the experimenter.

4 Conclusion

In this paper we present the results of a pilot study that seeks to establish how children react to a robot in different scenarios. Children with a higher level of social competence were assigned to an interaction scenario with the robot in which the social behaviors that accompanied the game were more complex. The children were rated as most happy and with a high arousal in level three and five. In level two and four the children were observed as the most bored. Mann and Whitney [17] stated that the emotion boredom can be classified with a low arousal and valence and the emotion happy can be classified with a high arousal and valence, which is consistent with our results and thus verifies the reliability of the observations.

The results indicate that the more complex the child robot interaction on a level of social competence is, the more engaged the children are. A possible alternative explanation of this result is, that the levels two and four were not challenging enough (not well enough designed) for the children or that they were misclassified. However, if the levels were not challenging enough for the children it does not explain why the children in game level three are observed differently. Game level three expects more social interactions of the children than game level two, however less than in game level four.

In level four and five the robot asked sometimes on purpose for the wrong card in order to initiate protesting behavior. This resulted in the child looking at the therapist with confusion. When the robot answered the child’s confusion with an explanation that he asked for the wrong card, the child had more attention to the robot afterwards. When the therapist answered that the robot asked the wrong card and the robot also said this after the therapist, the child kept asking questions to the therapist.

As Robins and Dautenhahn [19] suggested, the teacher or therapist in the room should be an active part of the interaction between child and robot. In this study the therapist was used as an assistant of the robot and this presence was needed for responding to each unexpected initiation of the child. When the robot could not grab his cards, the robot asked the therapist for help. In this pilot, a triadic interaction between teacher, robot and child is used. However, in a previous study of Barakova et al. [1] a triadic interaction was used between the robot and two children. The authors found that the children appreciated the personal attention of the robot. Therefore, in the current experimental setting the therapist is an assistant of the robot instead of part of the main interaction.

To further improve the design of the experiment the therapist should interfere less with the child. It is possible to pre-program a few explanation behaviors for the robot beforehand to execute during the game, since text-to speech explanation is slow and influences the interaction. More engagement between the robot and a child results in improvement of the communication skills of the child [7]. A faster response of the robot can result in higher engagement of the child with the robot and therefore in more effective improvement of his/her communication skills. Moreover, the robot used a computer-generated laughter, in a next experiment this can be changed into a recorded (more natural) laugher.

Extensive research has been done towards research with children and robots in a Wizard of Oz setting (see [15]) with the researcher being invisible for the child. However, less research has been done with the experimenter in the same room, which in some educational settings is preferable, for instance in therapy with children with ASD. In this case the therapist stayed in the same room to observe how the child was interacting with the robot and this will have an influence on the robot-child interaction, mostly favoring the human-human interaction. The current study is a preparation for such an experiment and the social competence levels are designed for the children with ASD. However, the description of the levels of social competence are general enough can be useful for enhancing the engagement of a general education, for instance of language and social skills training.

References

1. Barakova, E.I., Bajracharya, P., Willemsen, M., Lourens, T., Huskens, B.: Long-term LEGO therapy with humanoid robot for children with ASD. Expert Syst. **32**(6), 698–709 (2015)
2. Kose-Bagci, H., Ferrari, E., Dautenhahn, K., Syrdal, D.S., Nehaniv, C.L.: Effects of embodiment and gestures on social interaction in drumming games with a humanoid robot. Adv. Robot. **23**(14), 1951–1996 (2009)
3. Diehl, J.J., Crowell, C.R., Villano, M., Wier, K., Tang, K., Riek, L.D.: Clinical applications of robots in autism spectrum disorder diagnosis and treatment. In: Comprehensive Guide to Autism, pp. 411–422 (2014)
4. Gruarin, A., Westenberg, M.A., Barakova, E.I.: StepByStep: Design of an Interactive Pictorial Activity Game for Teaching Generalization Skills to Children with Autism. In: Entertainment Computing–ICEC 2013, pp. 87–92 (2013)
5. Huskens, B., Verschuur, R., Gillesen, J., Didden, R., Barakova, E.I.: Promoting question-asking in school-aged children with autism spectrum disorders: Effectiveness of a robot intervention compared to a human-trainer intervention. J. Dev. Neurorehabil. **16**(5), 345–356 (2013)
6. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). Biometrika **52**(3–4), 591–611 (1965)
7. Feil-Seifer, D., Mataric, M.: Robot-assisted therapy for children with autism spectrum disorders. In: Proceedings of the 7th International Conference on Interaction Design and Children, pp. 49–52 (2008)
8. Barakova, E.I., Gorbunov, R., Rauterberg, M.: Automatic interpretation of affective facial expressions in the context of interpersonal interaction. IEEE Trans. Hum.-Mach. Syst. **45**(4), 409–418 (2015)
9. Fujimoto, I., Matsumoto, T., Ravindra, P., Silva, S.D., Kobayashi, M., Higashi, M.: Mimicking and evaluating human motion to improve the imitation skill of children with autism through a robot. Int. J. Soc. Robot. **3**, 349–357 (2011)

10. Vanderborght, B., Simut, R., Saldien, J., Pop, C., Rusu, A.S., Pintea, S., David, D.O.: Using the social robot probo as a social story telling agent for children with ASD. Interact. Stud. **13**(3), 348–372 (2012)
11. Mehrabian, A.: Basic Dimensions for a General Psychological Theory Implications for Personality, Social, Environmental, and Developmental Studies (1980)
12. Lourens, T., Barakova, E.I.: User-Friendly Robot Environment for Creation of Social Scenarios. In: Foundations on Natural and Artificial Computation, pp. 212–221 (2011)
13. Ekman, P.: An argument for basic emotions. Cogn. Emot. **6**(3–4), 169–200 (1992)
14. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences. Lawrence Erlbaum Associates, Inc. (1977)
15. Riek, L.D.: Wizard of oz studies in HRI: a systematic review and new reporting guidelines. J. Hum. Robot Interact. **1**(1), 119–136 (2012)
16. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. J. Am. Stat. Assoc. **47**(260), 583–621 (1952)
17. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 50–60 (1947)
18. Byrt, T., Bishop, J., Carlin, J.B.: Bias, prevalence and kappa. J. Clin. Epidemiol. **46**(5), 423–429 (1993)
19. Robins, B., Dautenhahn, K.: The role of the experimenter in HRI research-a case study evaluation of children with autism interacting with a robotic toy. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2006, pp. 646–651 (2006)

Robot as Tutee

Lena Pareto

Abstract This paper explores the possible advantages of substituting teachable agents in a learning environment, with a humanoid robot as the non-human tutee. Teachable agents are used as an extension to educational games in order to leverage engagement, reflection and learning. The learning environment is engaging and shown to be effective for learning and promote self-efficacy in experimental studies in authentic classroom settings. Features beneficial for learning which are further enhanced by a robot compared to an agent are identified. These include embodiment of the robot; a social, empathic behaviour, better conversational abilities which together provide a better role model of an ideal learner for the student to identify with.

Keywords Robotics · Tutoring · Teachable agent · Robot tutee · Role-model learner

1 Introduction

Robots are starting to enter the classrooms, not as substitutes for human actors but as teaching tools or teaching assistants [1, 2]. According to [2], robotics is mostly used in computer science education, in domain-specific subjects such as geometry or physics where movement and spatial cognition are involved, in foreign language learning or as assistive tools for cognitive or social support, for example as storytelling assistant in pre-school [3]. Most applications are devoted to the engineering field not to the wider scope as a general educational tool [1]. However, in these classes students also learn general 21st century skills such as communication, collaboration, creative thinking, and problem-solving skills [4, 5].

Robots can play various roles in education; from design material in engineering to virtual companions in other learning situations. Our concept belongs to a learning

---

L. Pareto (✉)
University West, Media and Design, Trollhättan, Sweden
e-mail: lena.pareto@hv.se

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_24

situation where the human acts as a teacher/tutor to the virtual companion but where the purpose of the virtual companion is to act as a productive counterpart for the human learning [6]. There is a distinction between virtual companions that can be taught and those that can learn: to be teachable it is enough to appear to learn from the tutor, they do not need to 'learn' [7]. To simulate a tutee there are two objectives: to model a behavior that appears to progress due to teaching activities and that is understandable to the human tutor in order to monitor the teaching activity. In contrast to [8] where a robot learns social behavior from a child by machine learning, our robot tutee merely appears to learn and the focus is on features that support the human learner in becoming more competent.

The aim of this paper is to propose the Student Tutor And Robot Tutee (START) learning concept, based on a virtual tutee enhanced mathematical game, which is shown effective for learning conceptual understanding and reasoning in authentic classroom situations [9, 10]. There, teachable agents are used as an extension to educational games in order to leverage engagement, reflection and learning. Here, we explore robots as alternative to agents, and address the following question: *Can a robot tutee enhance learning experiences and increase learning effects compared to a virtual tutee such as a teachable agent?*

The paper describes features of a teachable agent game that are beneficial for learning, motivated by learning theories and experimental classroom studies. An analysis based on learning theories identifies features that can be further enhanced by migrating the tutee to a humanoid robot compared to the virtual teachable agent. The fields of robotics and artificial intelligence in education are thus combined [11].

2 Questions and Learning

Questioning is fundamental to learning and education, in particular for promoting deep-level reasoning of complex material. Questioning is an ordered event of three stages: an incentive of action; articulation; and a response action [12]. An ideal learner is an active, self-motivated, creative, inquisitive person who asks and searches for answers to deep questions [13]. Deep questions (such as why, why not, how, what-if, what-if-not) target difficult scientific material, whereas shallow questions (who, what, when, where) more often target simple facts.

There is substantial evidence that human tutoring is effective [14]. Tutees learn from peer tutoring; tutors gain in understanding as well [15]. Typically, the tutor is untrained and the knowledge gap between tutor and tutee can be minimal [15]. The tutor benefits from preparing material and reviewing their own understanding, and trying to respond to tutee questions or confusion if explanations are inadequate, incomplete or contradictory. Such tutoring effects make grounds for the learning-by-teaching paradigm and thus teachable agents [16] and virtual tutees in general.

3 The Learning Environment

We are currently using teachable agents as an extension to educational games in order to leverage engagement, reflection and learning. Teachable agents are computer-simulated peers that “learn” based on input from the student. Artificial intelligence techniques guide the agent’s behaviour based on what it was taught and students can revise their agents’ knowledge (and their own) based on such behaviour.

In most teachable agent systems, the pedagogical model is that of direct instruction. Students teach their agents by representing knowledge in a pre-determined format such as a concept map. This presumes that the student-as-teacher is able to articulate her knowledge in the required format. Such an assumption may not be appropriate when targeting younger students and domains such as causal reasoning and mathematical understanding, and we have taken a different approach: students teach their agents by demonstrating their game playing know-how and the agent acts “the ideal learner” and poses numerous of deep, explanatory questions prompting the student to reflect on and explain her playing behaviour. The purpose of the teachable agent dialogue is to challenge student tutors to reflect on their actions; to support transfer of tacit game playing skills to verbal language; and to provide a role-model of the ideal learner to imitate.

The teachable agent is designed as an inquisitive, active learner that takes the initiative and challenges the student to explain and justify her actions and thinking. Successful learners use self-explanation as a learning strategy [15]. Most students do not spontaneously self-explain but they do so when prompted and can learn to do it effectively. The level of difficulty in the agent’s questions follows the game performance level of the human tutor to challenge the student without being de-motivating.

The tutee learns from observing the tutors’ playing-behaviors and from the tutor’s responses to questions. Good as well as less good game playing behavior is considered. Similarly, having robots learn from positive as well as negative examples in [8] created better behavior. The tutee has a representation of the domain knowledge, which is hidden until the tutor demonstrates the corresponding knowledge, and that way the tutee appears to learn. The tutee’s question asking behavior follows a progression tree; if the tutor does not manage to answer questions at one level, it will not proceed with more advanced questions. This way, the agent will not become more competent than the tutor.

A teachable robot, claimed to be the only one, is a care-receiving robot that promotes language learning for young children [17]. It is taught by direct instruction by moving the robot, telling the robot what to do, or by showing an action. Our tutee is taught indirectly by observing the tutor act and asking questions based on the tutor’s actions, under the pretext that the tutee wants to understand. That way it plays the role of a curious, inquisitive learner, eager to understand. It is an active role where the tutee and the human tutor interact in a dialogue around the learning topic with the purpose to stimulate the student tutor to reflect on and explicate his/her actions. This process resembles scientific inquiry, since the tutee asks deep why-questions related to the learning material and the reasoning becomes explicit and visible. Hence, the learning process takes place in a social constructive environment [18].

The learning environment has shown to yield significant learning gain for playing students compared to controls, it engages children in advanced mathematical thinking in early education, and young primary students can act as successful tutors [10]. According to [1], few studies support their claim of effectiveness with quantitative evidence. This idea combines the motivational power of games with the reflective power of a virtual tutee asking thought-provoking, deep questions on the learning material during game play.

4 Features of Student as Tutor

The learning is based on a joint, engaging activity, where the tutor and tutee have a task to perform together. It is the interaction between the tutor and the tutee that further leverage the learning already designed in the activity. It must be a meaningful activity with an explicit learning goal, and games are good candidates. Assigning the role as tutor to students have the following features related to learning.

The students are assigned the role as tutor, but since the virtual tutee asks insightful questions and prompts plausible explanations they might not accept the fiction. However, 92 % of the students claimed that they taught the agent, and not vice versa [10]. Such fiction adoption is more likely to occur if the tutee behaves in a way natural to the tutor [19]. By natural we mean that (1) the questions should be of the form that the students could have asked themselves, (2) the timing of the question should be reasonable, and 3) the tutee should not become too clever too soon.

To teach the agent was ranked as the most engaging activity, compared to watching the agent play or play the game without the agent [10]. Still, the agent was not visually present besides as a small face image on the screen, so the idea of the teachable agent was enough to stimulate motivation. Collaborating with a humanoid robot compared to a simple image and the idea of an agent ought to enhance motivation a great deal.

The learning environment improved students' self-efficacy beliefs [9] compared to students not playing the game. An explanation can be that the tutee was ignorant from start and tutors who feel more capable exert more effort toward tutoring [15]. Also, novice peer tutors can feel anxiety about tutoring a human peer, but when the tutee is a computer agent such responsibility is removed. The tutee introduces the so-called protégé effect, i.e., an ego-protective buffer, since it is the tutee's knowledge that is in focus instead of the student's [20].

Features beneficial for student learning include students acting in an expert role, a role seldom used in education, but is a method lauded among learning theorists. Taking on a role or identity is one of the most effective ways of learning to think in new ways and learn new subject matter [18]. The teaching activity is known to be beneficial, and having to respond to the tutee's reflective questions ought to enhance learning, since question-driven explanatory reasoning appears to be the primary factor that explains why one-to-one tutoring is one of the most effective methods of learning a body of knowledge or a skill [16].

Finally, the tutor needs to evaluate and judge the tutee's behaviour and performance in order to teach, as well as negotiating and reasoning with the tutee who ensures that the conversation remains around domain-relevant topics.

5 Features of Robot as Tutee

The virtual companion is assigned the role as tutee, which become a genuine situation compared to when teachers asks questions. Teachers' questioning is not genuine [16]: they are not interested in the answer to learn, rather to judge the student's knowledge. The robot will be programmed to act as a learner, ignorant at start and behave as if it learns by observing the tutor's action and responses to questions. A low competency pedagogical agent is more motivational than a high-competency agent [21].

The embodiment and social behaviour of a robot makes the collaboration and the dialogue more believable compared to the teachable agent. There is evidence from neuroscience that the more human-like technology appears; the easier it is to accept it having intelligent features [11], and human-like robots are most believable after humans. The tutor-tutee dialogue is highly situational and interactional: the tutee robot reacts in direct response to student tutor's actions. More human-like actions from a social robot ought to enhance motivation.

Features beneficial for student learning where a virtual companion actually beats a human peer, concern directing attention to pre-defined learning issues and staying on the topic. Moreover, such behaviour is accepted *since virtual companions need not be social.* The virtual tutee acts according to its knowledge, which is a reflection on the tutors observed and explicit knowledge. Hence it's behaviour is related to the interacting partner, and could be personalized according the student's learning style or preferences [3], or to learners' special needs such as children with autism [22]. Also, this makes the tutee ask questions within the tutor's zone of proximal development [18], which cannot be assured with human collaboration.

The student interacts with the tutee and constructs knowledge through dialogue, an approach argued for in [6]. The dialogue is essential, and can be controlled since the robot is pre-programmed for the intended type of dialogue and topic. Hence, our approach is similar to [6], where students can negotiate their ideas with a humanoid robot and learn by means of socio-cognitive conflicts. Their study indicates that the robot-child dialogue was more effective than the human-child counter part. Their results are promising despite the small sample size and a novelty-effect of robots.

The tutee's thought-provoking questions encourage the tutor to self-explain, i.e., when a learner asks herself deep, explanatory questions and searches for answers. A robot ought to act an ideal learner more convincingly than an abstract teachable agent, and thus provide a better role model for the tutor to identify with.

6 Conclusion and Future Work

The START concept where a virtual companion is migrated from a teachable agent to a robot tutee, is argued to further enhance the learning situation due to (1) the embodiment of the robot; (2) a social, empathic behaviour (eye gaze, facial expressions, gestures) possible to implement in the robot, (3) better conversational abilities which all together provide a better role model of an ideal learner for the student to identify with.

Future work includes setting up a Wizard-of-Oz experiment with the same dialogue protocol as in the teachable agent-based learning environment, with a social, humanoid robot.

References

1. Benitti, F.B.V.: Exploring the educational potential of robotics in schools: a systematic review. Comput. Educ. **58**(3), 978–988 (2012)
2. Mubin, O., Stevens, C.J., Shahid, S., Al Mahmud, A., Dong, JJ.: A review of the applicability of robots in education. J. Tech. Educ. Learn. **1**, 209-0015 (2013)
3. Fridin, M.: Storytelling by a kindergarten social assistive robot: a tool for constructive learning in preschool education. Comput. Educ. **70**, 53–64 (2014)
4. Eguchi, A.: Educational robotics for promoting 21st century skills. J. Autom. Mobile Robot. Intell. Syst. **8**(1), 5–11 (2014)
5. Alimisis, D.: Robotics in education & education in robotics: Shifting focus from technology to pedagogy. In Proceedings of the 3rd International Conference on Robotics in Education, pp. 7–14 (2012)
6. Mazzoni, E., Benvenuti, M.: A robot-partner for preschool children learning english using socio-cognitive conflict. Educ. Tech. Soc. **18**(4), 474–485 (2015)
7. Brophy, S., Biswas, G., Katzlberger, T., Bransford, J., Schwartz, D.: Teachable agents: combining insights from learning theory and computer science. In: Lajoie, S.P., Vivet, M. (eds.) Artificial Intelligence in Education, pp. 21–28. IOS Press, Amsterdam (1999)
8. Cakmak, M., DePalma, N., Arriaga, R.I., Thomaz, A.L.: Exploiting social partners in robot learning. Auton. Robot. **29**(3–4), 309–329 (2010)
9. Pareto, L., Arvemo, T., Dahl, Y., Haake, M., Gulz, A.: A Teachable-agent arithmetic game's effects on mathematics understanding, attitude and self-efficacy. In: Biswas, G., Bull, S., Kay, J., Mitrovic, A. (eds.) Proceedings of the International Conference on Artificial Intelligence in Education, pp. 247–255. Springer, Heidelberg, Germany (2011)
10. Pareto, L.: A teachable agent game engaging primary school children to learn arithmetic concepts and reasoning. Int. J. Artif. Intell. Educ. **24**(3), 251–283
11. Timms, M.J.: Letting artificial intelligence in education out of the box: educational cobots and smart classrooms. Int. J. Artif. Intell. Educ. 1–12 (2016)

12. Van der Meij, H.: Student questioning: a componential analysis. Learn. Individ. Differ. **6**, 137–161 (1994)
13. Graesser, A.C., Person, N.K.: Question asking during tutoring. Am. Educ. Res. J. **31**, 104–137 (1994)
14. Craig, S.D., Sullins, J., Witherspoon, A., Gholson, B.: Deep-level reasoning questions effect: the role of dialog and deep-level reasoning questions during vicarious learning. Cogn. Instr. **24** (4), 563–589 (2006)
15. Roscoe, R.D., Chi, M.T.: Understanding tutor learning: knowledge-building and knowledge-telling in peer tutors' explanations and questions. Rev. Educ. Res. **77**(4), 534–574 (2007)
16. Biswas, G., Katzlberger, T., Brandford, J., Schwartz D.L.: TAG-V.: extending intelligent learning environments with teachable agents to enhance learning. In: J.D. Moore, Redfield, C. L., Johnson, W.L. (eds.) Artificial Intelligence in Education, pp. 389–397 (2001)
17. Tanaka, F., Matsuzoe, S.: Children teach a care-receiving robot to promote their learning: field experiments in a classroom for vocabulary learning. J. Human Robot Interact. **1**(1) (2012)
18. Vygotsky, L.: Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge, MA (1978)
19. Chan, T.-W., Chou, C.-Y.: Exploring the design of computer supports for reciprocal tutoring. Int. J. Artif. Intell. Educ. **8**, 1–29 (1997)
20. Chase, C., Chin, D.B., Oppezzo, M., Schwartz, D.L.: Teachable agents and the protégé effect: increasing the effort towards learning. J. Sci. Educ. Technol. **18**(4), 334–352 (2009)
21. Kim, Y., Baylor, A.L.: Pedagogical agents as learning companions: the role of agent competency and type of interaction. Educ. Technol. Res. Dev. **54**(3), 223–243 (2006)
22. Weiss, P.L., Cobb, S.V.G., Zancanaro, M.: Challenges in developing new technologies for special needs education: a force-field analysis. In: 10th International Conference on Disability, Virtual Reality and Associated Technologies, Sweden (2014)

Concept Inventories for Quality Assurance of Study Programs in Robotics

Reinhard Gerndt and Jens Lüssem

Abstract Robotics is gaining an importance among the subjects to be studied in scientific and engineering disciplines. However, being a quite new discipline with a high degree of inter- and trans-disciplinary aspects, the teaching community cannot rely on experience, gained by a long history of experiments. With this paper we propose means to help assess and improve study programs in robotics. The approach is based on the idea of concept inventories (CIs) and related tests to derive objective information for a comparison of student groups with each other and over time, e.g. to assess learning gain for specific measures. The approach helps to further establish quality assurance in the design of robotics study programs and equips teachers with measures for a formative assessment of their work.

Keywords Robotics education • Concept inventory • Teaching assessment • Learning assessment • Quality assurance • Accreditation

1 Introduction

Quality assurance as a dedicated task has been introduced to many universities with the Bologna process, targeted to harmonizing the European Higher Education Area.

There are a number of organisations working on the definition of standards and guidelines. As a representative for others, [1] requests “Institutions should have a policy and associated procedures for the assurance of the quality and standards of their programmes …”. However, the requirements typically are not broken down to the operational level. Moreover, often, the quality assurance procedure is based on

---

R. Gerndt
Ostfalia University, Wolfenbuettel, Germany
e-mail: r.gerndt@ostfalia.de

J. Lüssem (✉)
University of Applied Sciences Kiel, Kiel, Germany
e-mail: jens.luessem@fh-kiel.de

M. Merdan et al. (eds.), *Robotics in Education*, Advances in Intelligent Systems and Computing 457, DOI 10.1007/978-3-319-42975-5_25

review by independent external agencies, either at the level of individual study programs or at system level.

This introduces a subjective element to the process, which may result in a unification of programs and hamper development of individual fields of expertise and excellence. Whilst definition of standards is a commendable objective of these activities, they often lack sufficient objective components, e.g. a formative assessment of students and knowledge gain.

Currently, quality assurance often is biased by personal experience of the, internal or external, auditors or members of advisory boards or by a given industrial or research community involved. However, this only insufficiently allows for a more agile reaction to new requirements by industry, service providers and academia.

To overcome these limitations, new means of quality assurance for study programs should therefore include both, organisational processes and formative methods.

The remaining part of this paper is organized as follows: After this introduction we will shortly revisit the organisational measures for quality assurance in higher education. Then we will add some notes on concept inventories as means of quality assurance. Section 4 will specifically address the robotics concept inventory and present experimental data on its application. Eventually, we sketch how organisational means and concept inventories can be combined for a long-term management of study programs and present our conclusions.

2 Quality Assurance of Study Programs

Since the start of the Bologna process, higher education institutions have established internal quality management systems for their study programs.

Quality assurance agencies had taken the role of external auditors. Initially, quality assurance agencies audited study programs. Thus, the—internal and external—evaluation of study programmes was in the centre of interest (see Fig. 1).

Since ten to fifteen years, quality assurance agencies have moved towards an institutional audit approach to quality. Thus, higher education institutes have started to build organisation-wide quality assurance systems.

For at least a decade, we see a growing quality assurance community within higher education institutions. Networks are growing across Europe.

Today, a number of higher education institutions with mature quality management systems have already moved away from programme accreditations to a so called system accreditation. An example for a quality assurance system can be taken from Fig. 2.

The evaluation of courses and the curricula seem to be still in the centre of interest (see Fig. 2), but higher education institutions focused more and more on their internal processes—in preparation for external audits.

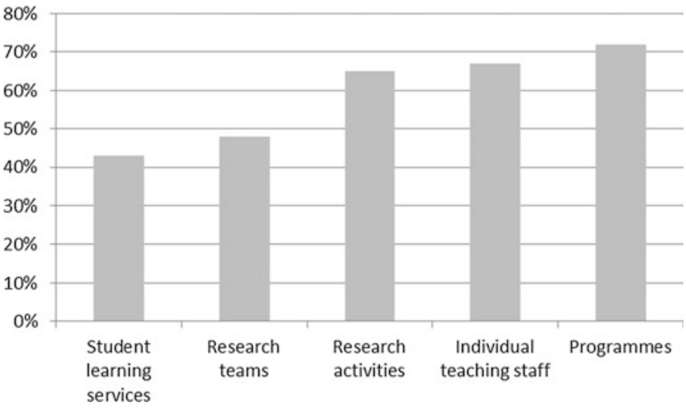


Fig. 1 Regular internal evaluation (from [2])

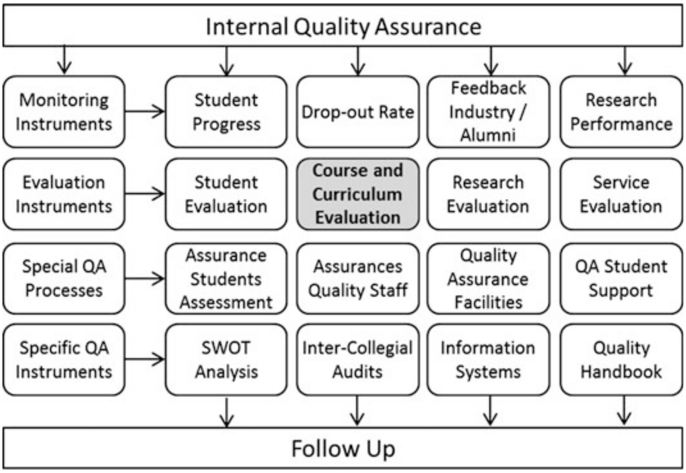


Fig. 2 Model for a quality assurance system (adapted from [3])

Thus, higher education institutions shifted away little by little from the evaluation of the content of their study programs.

The reasons for this shift are manifold:

- Quality assurance agencies require sound quality management systems—the content of a study program or a course is only a part of these management systems;
- Higher education institutions focus on processes to meet these requirements;
- Higher education institutions use evaluation criteria that can be used on an organisational level—this tends to exclude evaluation criteria related to the content of a study program or a course

- Study program managers have not the right means to evaluate courses or curricula.

We think that concept inventories can be very helpful to refocus institutions to the content dimension.

3 Concept Inventories as a Means of Quality Assurance

Concept inventories have been developed in order to determine whether a student has a working knowledge of a specific set of concepts. Concept inventories are designed and evaluated to ensure test reliability and validity.

Therefore, in our opinion, concept inventories can be a great help for study program managers to evaluate courses, the linkage of courses or even whole study programs.

Figure 3 shows a crucial part of a (typical) conceptual scheme for curriculum development and evaluation: Expected learning outcomes may differ considerably from the real learning outcomes. Concept inventories may be a reliable instrument to determine the real learning outcomes.

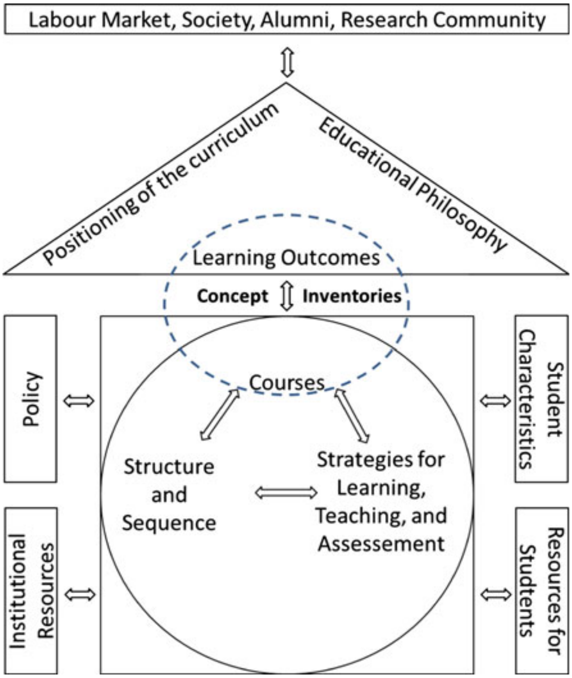


Fig. 3 Conceptual scheme for curriculum development and evaluation (adapted from [4])

4 Overview Over the Robotics Concept Inventory

Concept inventories intend to list the relevant concepts that are required to master a specific scientific field. With every concept inventory there is a respective test that allows assessing the understanding of students of the relevant concepts independently of the actual knowledge. Students typically undergo the test twice, first as a ‘pre-test’ at the beginning of the course and second as a ‘post-test’ at the end.

Since tests do not change over some time or only develop slowly, results can be used to assess the overall input level of students within their peer group. Applying the test a second time, after attending a specific course, allows the assessment of the concept learning gain.

A series of similar test methods has been developed, i.e. [5]. Because of the concept learning gain we think that concept inventories can be integrated a bit easier in our quality assurance system.

Figure 4 visualizes the gains for a number of teaching approaches in “Signals and Systems” courses [6].

A number of CIs have been developed for different fields of studies (e.g. [7, 4]) one of the first CIs with a direct relation to our robotic CI is the force concept inventory ([8]).

Our specific robotics CI has been presented in [9]. The relevant concept classes —derived from textbooks, curricula, and course syllabi (e.g. [8, 10–13])—are listed here:

- Math/Numerical Methods
- Mechanics
- Control Theory
- Stability
- Kinematics
- Dynamics

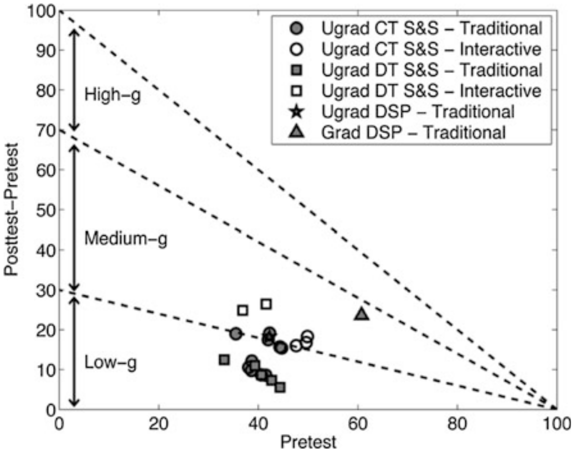


Fig. 4 Averaged gains of different course types (from [6])

- Sensing
- Perception
- Planning
- Navigation
- Decision-making
- Uncertainty

The partial robotics CI presented in [9] has been verified with pre- and post-test at 50 % of the course time in a robotics course in the year 2015 and with a pre-test only in 2016. Both courses took place at the same university, with master students with comparable background. In 2015, almost 30 students participated in the pre-test only, of whom 8 students also took part in the post-test, whilst up to the time of writing this paper in 2016, 14 participated in the pre-test only. The numbers are small and thus conclusions need to be drawn with care.

4.1 Evaluating the Learning Gain

The formula to evaluate the learning gain as visualized in (Fig. 4) is calculated by the following formula:

$$gain = \frac{post - pre}{100 - pre}$$

Results showed a value of *pre = 40 %* for the percentage of positive answers in the pre-test and *post = 53 %* for the percentage of the positive answers in the (early) post-test. Applied to the formula, the gain can be computed to:

$$gain = \frac{53 - 40}{100 - 40} = 0.22$$

Comparing this learning gain to the results for the Signals and System CI (Fig. 4), following information can be derived:

(1) A pre-test performance of roughly 40 % would make the student group comparable to undergraduate students. However, the Robotics CI still lacks sufficient verification and calibration, such that a classification with respect to the entry level of students may be too early. However, there is some credibility to the result, since many students came from the software domain, possibly making them comparable to undergraduate Robotics students.

(2) A post-test performance of 53 % results in a gain of roughly 0.21, which groups the course with low-gain courses. However, since the 'post'-test was taken at about half time of the course (after having finished the theoretical part), further improvement was to be expected. This would have moved the course at least to the border of low and medium gain courses.

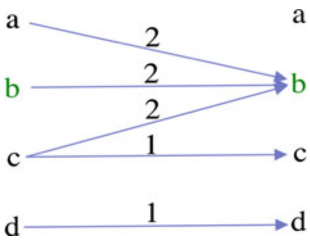


Fig. 5 Answering behaviour: transformation question

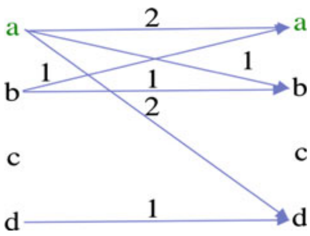


Fig. 6 Answering behavior: rigid body mechanisms

Significantly more interesting from a lecturer's point of view was the individual answering behaviour. This can be illustrated with two examples.

(1) The changes between individual answers for a specific question related with transformations are shown in Fig. 5. The left column indicates the answers given at the pre-test, whilst the right column indicates the answers given at the post-test. In this case, answer b was the correct one. The arrows indicate how answering behaviour of individual students changed and the respective numbers. This example clearly shows convergence towards the correct answer, which can be considered an outcome of the course.

(2) The changes between individual answers for a specific question related with rigid body mechanics are shown in Fig. 6. In this case, answer "a" was the correct one. This example shows how many student, after answering the question correctly in the pre-test, got detracted and picked a wrong answer in the post-test. This kind of 'unlearning' may be a necessary step for students to overcome incorrect concepts, which may be the case here, with the test taken at half time, or it may indicate an unfavourable teaching approach.

4.2 Comparing Student Groups

A single CI (pre-) test may also allow comparing student groups with respect to their state when entering the course. The information may be helpful to focus course work on relevant aspects. However, comparing results from different students with comparable developments also allows calibration of the CI tests.

Table 1 Percentage of correct answers

#	Question	Correct answers 2015 (%)	Correct answers 2016 (%)
1	Transformation	34	36
2	Time shift	52	71
3	Acceleration[a]	52	71
4	Small-signal/linearization	55	50
5	Mass-damper	21	7
6	Segway	86	79
7	Maze	28	7
8	M-Bot	52	50

[a] Text of question was changed to avoid potential ambiguity

First we compared the percentage of students of the group who selected the correct answer in the pre-test of the 2015 and 2016 student groups. The results are shown in Table 1.

The comparison of results between 2015 and 2016 student answers shows quite the same results for questions number 1, 4, 6 and 8. In summary, at the pre-test in 2015 correct answers were given with an average of roughly 47 % and a standard deviation of 20 %. In 2016 the average for correct answers reached the same value, however the standard deviation of 28 % was larger than in the year before.

Next we compared results with respect to the majority of the answers that were assumed to be correct (Table 2).

This table shows quite the same results with respect to the remaining questions 2, 3, 5 and 7. In all cases a majority consistently considered a right, respectively wrong answer to be correct one. However, qualitatively the latter ones show a significant deviation in the range of 14 to 21 points on the percentage scale (which relates to 2–3 students in the 2016 test). Noteworthy to mention is that question 2 (time shift), which shows a significant deviation of results, is taken from the mature and well calibrated Signal and Systems concept inventory [6].

Table 2 Majority of answers

#	Question	Majority of answers 2015 (%)	Majority of answers 2016 (%)
1	Transformation	34 (correct)	36 (correct)
2	Time shift	52 (correct)	71 (correct)
3	Acceleration[a]	52 (correct)	71 (correct)
4	Small-signal/linearization	55 (correct)	50 (correct)
5	Mass-damper	45 (wrong)	71 (wrong)
6	Segway	86 (correct)	79 (correct)
7	Maze	55 (wrong)	79 (wrong)
8	M-Bot	52 (correct)	50 (correct)

[a] Text of question was changed to avoid potential ambiguity

With respect to the number of correct answers of individual students in the pre-test, the following numbers have been recorded: In 2015 the average number of correct answers was 3.25 out of eight with a standard deviation of 1.3. Average number for 2016 has been 3.7 with a standard deviation of 1.8. This indicates a slightly higher level of expertise in the 2016 student group, however, it also indicates a wider diversity compared with the 2015 student group. Results can be compared to an average number of correct answers of the respective post-test of 4.25 with a standard deviation of 1.5. These numbers indicate an overall gain of knowledge in the 2015 group through the lecture. However, numbers also indicate that the spread between students increased. This is reflected by the stronger deviation of correct answers for individual questions in the 2015 pre-test.

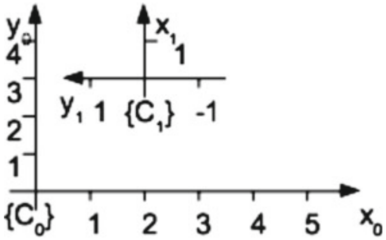
Aside from aggregated information from all questions, the individual answers provide further information. We will provide some details on question 1 of the Robotics CI (topic: transformation), as shown in Fig. 7.

With 34 % and 36 % choosing the correct answer in 2015 resp. 2016, 76 % and 74 % considered wrong answers to be correct (Table 3). Furthermore, the answers initially are quite equally distributed among answers "a" to "c", which shows no pre-occupation about this concept. Consequently, learning a suitable concept can possibly be achieved more easily.

Fig. 7 Question 1 of robotics CI

Question 1

Given coordinate systems C_0 and C_1 as in the following image:



For transformation of the coordinates of a point P from coordinate systems C_1 to C_0 we can find a system of linear equations. Identify the suitable equations to calculate x_0 and y_0 values of point P with its coordinates given in x_1 and y_1 values.

a) $x_0 = x_1 + 2$, $y_0 = y_1 + 3$,
b) $x_0 = - y_1 + 2$, $y_0 = x_1 + 3$,
c) $x_0 = - y_1 - 2$, $y_0 = x_1 - 3$,
d) $x_0 = 2y_1 -3$, $y_0 = -3x_1 + 2$.

Table 3 Distribution of answers for question 1 (Transformation)

Answers (%)	a	b	c	d	none
2015	31	34	21	3	10
2016	21	36	29	14	0

(numbers may not sum up to 100 due to rounding errors)

5 Management of a Study Program Using the Robotics Concept Inventory

So far, we have used the robotics concept inventory mainly

- to adjust our teaching methodology in our robotics courses;
- to change the structure of our robotics courses;
- to rearrange and adjust the content of our robotics courses;
- to redefine the “interfaces” between two courses.

But we think that it is even possible to work with a range of concept inventories in order to help evaluating a whole study program.

6 Conclusions and Outlook

In this paper we presented how definition of conception learning outcomes and formal assessment can be combined with organizational procedures to assure quality in higher education in the field of robotics. We provided positive indicators on the viability of the concept inventory approach. Moreover, we provided information on how the approach can be applied at different levels of quality assurance, from assessing and managing the learning outcome of individual students, over student groups to entire higher-education organizations.

References

1. European Association for Quality Assurance in Higher Education: Standards and Guidelines for Quality Assurance in the European Higher Education Area, Helsinki (2009). http://www.enqa.eu/pubs.lasso
2. Crosier, D., Purser, L., Smidt, H.: Trends V—Universities shaping the European higher education area. European University Association, Brussels (2007)
3. Vroeijenstijn, T.: A journey to uplift quality assurance in the ASEAN universities. Report of the AUNP (2006)
4. Ogunfunmi, T., Herman, G.L., Rahman, M.: On the use of concept inventories for circuit and systems courses. IEEE Circuit Syst. Mag. Third Quarter (2014)

5. Ahlgren, D. Verner, I.: 2006–2015: Robotics Olympiads: a new means to integrate theory and practice in robotics. In: Annual Conference, American Association for Engineering Education, Chicago (2006)
6. Wage, K.E., Buck, J.R., Wright, C.H.G., Welch, T.B.: The signal and systems concept inventory. IEEE Trans. Educ. **48**(3), 448–461 (2005)
7. Lindell, R.S., Peak, E., Foster, T.M.: Are they all created equal? A comparison of different concept inventory development methodologies. PERC Proc. **883**, 14–17 (2006)
8. Hestenes, D., Wells, M., Swackhamer, G.: Force Concept Inventory. The physics teacher (1992)
9. Gerndt, R., Lüssem, J.: Towards a robotics concept inventory. In: 6th International Conference on Robotics in Education. Yverdon-les-Bains, Switzerland (2015)
10. Featherstone, R.: Rigid Body Dynamics Algorithms. Springer (2008)
11. Kelly, A.: Mobile Robotics—Mathematics, Models and Methods. Cambridge University Press (2013)
12. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. The MIT Press, 2005
13. http://ocw.mit.edu/courses/mechanical-engineering/2-12-introduction-to-robotics-fall-2005/syllabus Accessed 12 Mar 2016
14. Totté, N., Huyghe, S. Verhagen, A.: Building the curriculum in higher education—a conceptual framework. In: Enhancement and Innovation in Higher Education, Glasgow, United Kingdom (2013)