REVIEW

Technologies for an inclusive robotics education [version 1; peer review: 2 approved with reservations]

Dimitris ALIMISIS

EDUMOTIVA (European Lab for Educational Technology), Kapnikareas 19A, Athens, 10556, Greece

Abstract

The H2020 project “INBOTS: Inclusive Robotics for a Better Society” (2018–21) has worked in different disciplines involved in the acceptance and uptake of interactive robotics, including the promotion of accessible and multidisciplinary education programs. In INBOTS, educational robotics is considered as a learning tool that can bring robotics into school classrooms and benefit all children regardless of their future educational or professional orientation. Aiming to make robotics education inclusive, INBOTS has introduced a paradigm shift inspired by sound pedagogies (Papert’s constructionism) and emerging educational trends (the maker movement) and focused on creativity and other 21st-century skills. However, the realisation of this new paradigm requires appropriate curricula and technologies at both hardware and software levels. This paper addresses several questions and dilemmas related to the technologies currently in use in robotics education and the kind of technologies that can best support the proposed paradigm. This discussion results in specific criteria that robotics technologies must fulfil to foster the new paradigm. Based on these criteria, we review some representative technologies in both hardware and software. Then, we identify and discuss some technological solutions that exemplify the kind of technologies that can best support inclusive robotics education and make the proposed paradigm feasible. Finally, we show how some of these technologies can be combined to design a creative and inclusive project consistent with the criteria set in this paper.

Keywords

learning technologies, robotics education, maker movement, inclusive education, INBOTS.

This article is included in the Industrial Leadership gateway.
Introduction: a new paradigm in educational robotics

More than ever before, educators are using robotics kits and educational robots to teach and inspire students of all ages, from pre-school to secondary school, preparing them for a society in which robotics are used more and more often in everyday life. Robotics education is usually conceived as a two-fold mission: education in robotics and education with robotics. The in case promotes knowledge and understanding of the design, analysis, application, and operation of robots; the with concept is broader and entails the use of robotics as a tool for teaching and learning science, technology, engineering, arts and math (STEAM) subjects and developing 21st-century skills: creativity, problem-solving, critical thinking and teamwork, for example 1, 2.

This distinction between the two branches of robotics education is important because it affects the inclusiveness of the project. Learners in school education are still shaping their interests in and attitudes to different school disciplines, including engineering, technology, and robotics. Only a few of them will pursue university studies or professional careers in robotics or engineering or become software developers. However, education with robotics promises benefits for all children, no matter their future educational or professional orientation 1. Is this great promise and potential of robotics education considered and respected in the design of new curricula and technologies before they are brought into classrooms?

The H2020 project “INBOTS: Inclusive Robotics for a Better Society” brings together experts from different disciplines involved in the acceptance and uptake of interactive robotics, including the promotion of accessible and multidisciplinary education programs. In INBOTS, educational robotics is considered as a learning tool that can bring robotics into school classrooms and benefit all children regardless of their future educational or professional orientation. The INBOTS education team has worked (2018–2021) to develop a sustainable framework that will promote inclusive robotics education at EU level, focusing on education with robotics for three age groups: pre-school (age 4–6), primary school (age 7–12) and secondary school (age 13–17).

To make robotics education inclusive of all children, the INBOTS interventions have introduced a paradigm shift inspired by sound pedagogies (Papert’s constructionism 1) and emerging educational trends (the maker movement in education 1). We have already outlined a paradigm shift in robotics education that integrates the maker culture and the do-it-yourself (DIY) ethos with a focus on creativity and the other 21st century skills: problem-solving, critical thinking and teamwork. We are aware that the realisation of a new paradigm of this kind depends on appropriate curricula and technologies at both hardware and software level.

In the INBOTS framework, we review the more prominent curricula and recommend changes to align them with the proposed paradigm. In addition, a set of specific curricula and resources for school education that exemplify the new paradigm are currently under development. The INBOTS curricula and resources have been piloted with teachers and children in courses in Athens (Autumn–Winter 2019); a short video from pilots is available on YouTube. The curricula and resources will become available online to help teachers and educators implement the proposed paradigm in their classes and labs and to inspire them to create their own curricula and resources.

The new paradigm – and the INBOTS curricula – needs support from appropriate technological tools. We provide a systematic review of the available educational robotics technologies that appear in the literature and a long list of available resources.

Following up on this work within INBOTS, the present paper explores a central question emerging from our previous work: what kind of technologies (at hardware and software levels) are appropriate to support the new paradigm and engage young people with an inclusive robotics education? To answer this question, we need first to answer the following sub-questions:

– What criteria should inform decisions about the potential of particular technologies to support an inclusive robotics education?
– To what extent can the technologies currently used support an inclusive robotics education?

The following section addresses these questions by examining some critical dilemmas regarding technologies used in robotics education. In the process, we identify criteria that robotics technologies must fulfil to foster the new paradigm. We subsequently use these criteria to review some representative technologies at both hardware and software levels. We then propose and discuss some exemplary technological solutions to demonstrate the kinds of technologies that support inclusive robotics education and make the proposed paradigm feasible. Finally, we show how some of these technologies can be combined to design a creative and inclusive project compatible with our paradigm.

Discussion

Before exploring the technologies currently used in robotics education and how well they serve our proposed paradigm, it is necessary to establish some criteria for this discussion. To this end, some important questions and dilemmas within robotics education are highlighted in the following sub-sections.

*Arts and crafts robots or LEGO® MINDSTORMS robots?* This challenging dilemma comes from the title of a comparative study in which students in one group were asked to build a robot from scratch using craft and recycled materials while those in a second group were asked to build a robot out of structured materials. Results showed that building a robot from scratch increased pupils’ knowledge and manual skills while building a robot with structured materials increased their awareness of the robotisation of machines.

In other papers we have reported upon a project realised in the frame of the eCraft2Learn action in which children aged
13–17 created DIY robotic automobiles from scratch using low-cost and recycled materials; this project aligns with the “make your own robots” concept and the maker movement mindset. In the frame of the same action, we have seen young students in several projects working enthusiastically with creative “arts and crafts” robotic artefacts.

We concur with researchers who argue for technologies such as “creative material” that can offer a design which provides hands-on material for children to manipulate, think with and act upon as creators. The “arts and crafts” model allows children to un-box, un-craft, deconstruct and reconstruct, triggering their curiosity and facilitating collaboration; it ensures transparency and visibility of tools and materials and helps children understand how their robot works, its boundaries and its uses.

Technologies designed for professionals and hobbyists or for learners? Though the answer to this question seems (and is) obvious from a learning perspective, it is a common scenario in robotics education for teachers and learners to struggle managing technologies designed mainly for other purposes and only secondarily for education, if at all. For instance, prototyping technologies designed for professionals or enthusiasts, such as Arduino-like microcontrollers, can, when introduced in classrooms or makerspaces for novices, cause difficulties and eventually frustration and discouragement because they presuppose a domain knowledge (electronics) and skills (such as soldering, circuiting, C coding etc.) that young students or novices are unlikely to possess. Moreover, the producers of technologies for professionals or hobbyists focus on doing the job fast and at a low cost. They do not care much (nor is it their role) to help users understand how the product works; this often results in “overdesigned” technologies which essentially hide how the product actually works.

We have already argued that the technologies we invite children to interact with should be designed in line with sound learning theories and constructionist/constructivist pedagogy. However, this premise alone cannot guarantee the design of appropriate tools. Very often, technologies for learning offer “less for more” compared to corresponding professional tools or are not reliable. Putting emphasis on making technologies educationally meaningful and engaging should not compromise their reliability or scientific accuracy.

Should technologies prepare students for engineering jobs or develop personal expression and skills? Decisions regarding which objectives to pursue in robotics education have evident consequences for the choice of technologies deployed. If the emphasis is on preparing students for engineering jobs, then technologies replicating or simulating professional tools may be the right choice; however, if we focus on personal expression and skills development for all, regardless of future careers, then technologies compatible with the above-mentioned “arts and crafts” model, which allow children to design and make their own robotic artefacts from scratch, seem the right choice.

Should more time be spent on technicalities or creative tasks? When students start working on their robotics projects, they need some initial training to learn to use the technology. The time required for this training is an important factor for student engagement. According to our experiences in pilots, children are impatient to see concrete results and feel frustrated or even disappointed when they encounter difficulties using the tools or when it takes too long to progress from technicalities to creative tasks. Moreover, the attention demanded by technical issues may compromise their interest and undermine their engagement in the project. Thus, to allot more project time to creative tasks and less to technicalities, we need technology that is easy for learners to understand and use without exposing them to an unnecessary level of technical detail.

Scenarios from everyday life or “missions to Mars”? Very often in robotics competitions (e.g. the World Robot Olympiad) or in the curricula and manuals proposed by the producers of commercial robotics kits, “exotic” scenarios and models of robots inspired by fiction movies or popular TV shows are proposed to excite children’s interest. Industries advertise spectacular robots ready for space adventures, which sound exciting but have nothing to do with children’s everyday lives. There is a risk that this sort of marketing will result in the mystification of robots and promote misconceptions about how robotics technologies work and the reasoning behind them.

Cost matters. Finally, the high cost of robotic technologies may be an inhibiting factor for schools, educators and families. Low-cost kits that can be combined with materials from everyday life and recycled components of toys can offer creative solutions, often providing better educational value than expensive, ready-made robots or kits that allow children to make only a limited number of predefined models.

From this discussion the following main criteria arise for the kind of technologies that can serve best our inclusive paradigm:

- enable students to build DIY robots from scratch.
- offer transparency of their underlying structure.
- are designed for learners and not for professionals or hobbyists.
- do not require prior knowledge of the technology from outside the classroom.
- emphasise personal expression and skills development for all instead of preparing students for engineering jobs.
- allow for more time to be spent on creative tasks and less on technicalities.
- demystify robots and help children understand how their robot works.
- are of low cost for schools.

How do the technologies currently in use perform against the criteria set in the previous section? In Table 1, three popular
Platforms (LEGO Mindstorms, Arduino and Micro:bit) are assessed against these criteria to inform a discussion of the kinds of technologies that might best serve the proposed paradigm.

The three platforms compared in Table 1 represent quite different approaches within educational robotics. LEGO Mindstorms offers modular assembly of robots based on the popular LEGO construction sets for creative building. The Arduino platform, like many other microcontrollers available for physical computing, is based on boards that can read an input and turn it into an output, activating a motor or turning on an LED; teachers and students use these to build simple or complex robotic artefacts at low cost. The BBC Micro:bit platform offers an accessible pathway into electronics and computing, combining ease of construction with the power of engineering. Educators and children can use sensors and actuators connected to their Micro:bit board to make robots and other digital artefacts.

Table 1. Comparing platforms popular in educational robotics against the criteria arising from the discussion.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>LEGO Mindstorms</th>
<th>Arduino</th>
<th>Micro:bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>enables students to build DIY robots from scratch.</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>offers transparency of their underlying structure.</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>is designed for learners and not for professionals or hobbyists.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>requires prior knowledge of the technology from outside the classroom.</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>emphasises personal expression and skills development for all instead of preparing students for engineering jobs.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>allows for more time to be spent on creative tasks and less on technicalities.</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>demystifies robots and helps children understand how their robot works.</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>is of low cost for schools.</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

As Table 1 shows, LEGO Mindstorms and Arduino platforms meet some, but not all, of the set criteria, while Micro:bit performs quite well in most categories. The popular commercial platform LEGO Mindstorms builds on the familiarity to children with LEGO bricks and offers (though at high cost) easy modular assembling of robots and abstraction of unnecessary technical details. However, it lags behind the other two platforms in terms of openness to manipulation by the learner, transparency of the underlying electronics and potential to become a design material.

Arduino platform, though designed primarily for professionals and hobbyists, offers more opportunities for learners to “make their own robots”, can easily become a design tool in the hands of learners, and provides many opportunities for learners’ own manipulations. However, Arduino requires prior knowledge of the technology from outside the classroom (e.g. electronics, soldering skills). Moreover, students usually have to spend more time on technicalities than with LEGO Mindstorms or Micro:bit and may be frustrated by the many unnecessary details to which they are expected to attend. We concur with Blickstein: “a toolkit that is supposed to help students learn robotics but makes them spend half of their time figuring out how a breadboard works, will not be able to live up to its goals”15.

The Micro:bit platform seems to excel in certain aspects when compared with LEGO Mindstorms and Arduino. It carries most of the advantages of Arduino without some of the shortcomings. The electronics in Micro:bit are comparatively more transparent, without exposing users to unnecessary details; less prior technical knowledge and skill is required. Micro:bit combines expressive electronics with arts and crafts, and it is a good example of an inclusive technology that enables children to make their own low-cost robotic artefacts to support inclusive education in robotics, in line with our discussion in this article.

The programming language used is another factor that differentiates robotics technologies. Both text coding and visual programming languages (VPLs) are used in educational robotics. Text coding (e.g. Arduino IDE) is synonymous with a technical language and a rigorous syntax, while VPLs offer commands in visual blocks which prevent syntax errors. Some programming platforms (e.g. MakeCode) incorporate both options: children can use programming blocks in a visual environment while watching their commands appear in a second window in Python or JavaScript, which familiarises them with text coding.

In Table 2, Table 3 and Table 4, three different programming solutions for each of three common tasks are listed and then discussed. These tasks (turn on the LED in Table 2, play a note in Table 3, rotate the servo motor in Table 4) are ones that might be involved in classic projects in which children make a light-house, a Theremin music box and a sunflower, respectively. The (partial) solutions come from programming platforms well known in educational robotics: Arduino IDE (as an example of text coding with commands written in C/C++ language)
Another problem we identify in the Arduino C commands (Table 2, Table 3 and Table 4) is the “relatively unforgiving”\(^{16}\) and rigorous syntax required. This poses unnecessary obstacles for young programmers, who must pay attention to details of syntax and manage frequent syntax errors. This issue relates to the discussion about the role of “boring” technicalities, such as difficult syntax and unfamiliar terms in the programming interface that students must use to create behaviours for their robots. In contrast, visual programming solutions relieve children from worries about syntax and replace technical terms with a more intuitive and familiar language\(^{17}\).

The maker movement has inspired a range of efforts to produce STEAM and robotics kits and toys. In this section, some indicative examples are reviewed to discover how well they may serve our paradigm for robotics education based on the analysis of Table 1, Table 2, Table 3 and Table 4. The review begins with technologies that largely target younger (primary school) learners and continues to those mainly intended for older learners (secondary school).

Cardboard and paperboard-based crafting have become a nearly ubiquitous entry point into making and “can provide an easily accessible pathway into electronics and computing combining two modes of making, the power of engineering with the expressiveness of craft”\(^{18}\). Chibitronics offers a friendly way for kids to learn, design and create their own electronics projects using paper circuits (Figure 1), arts and crafts (Figure 2) and friendly programming (Figure 3). Children can use the Chibi Chip board (Figure 1) to bring their projects to life by connecting switches and sensors to it. The materials used look like familiar craft supplies; for instance, strips of copper tape or other conductive material are used to connect the Chibi Chip board to the LED (or other) circuit in a loop (Figure 2). The focus in Chibitronics is on combining expressive electronics with arts and crafts, and it currently supports only a few sensors and actuators, which means fewer options for robotics projects than Arduino-like microcontrollers can offer. However, this is a good example of a technology that enables children to make simple, low-cost arts and crafts projects, in line with our discussion.

### Table 2. Comparison of three tools for programming the task “turn on the LED”.

<table>
<thead>
<tr>
<th>Programming language</th>
<th>Lighthouse project/task: turn on the LED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino C</td>
<td>digitalWrite(9, HIGH);</td>
</tr>
<tr>
<td>Mblock</td>
<td>digitalWrite(9, HIGH);</td>
</tr>
<tr>
<td>Open Roberta Lab</td>
<td>turn LED on lighthouse on</td>
</tr>
</tbody>
</table>

### Table 3. Comparison of three tools for programming the task “play note”.

<table>
<thead>
<tr>
<th>Programming language</th>
<th>Theremin project/task: play note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino C</td>
<td>tone(buzzer_buzzer, 300, 100)</td>
</tr>
<tr>
<td>Mblock</td>
<td>play pin with note C4 for 0.25 beats</td>
</tr>
<tr>
<td>Open Roberta Lab</td>
<td>play frequency Hz duration ms</td>
</tr>
</tbody>
</table>

### Table 4. Comparison of three tools for programming the task “rotate the servo motor”

<table>
<thead>
<tr>
<th>Programming language</th>
<th>Sunflower project/task: rotate the servo motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino C</td>
<td>servo_Sunflower.wrire(90)</td>
</tr>
<tr>
<td>Mblock</td>
<td>set servo pin position to LED</td>
</tr>
<tr>
<td>Open Roberta Lab</td>
<td>set servo motor 9090 Sunflower re to LED</td>
</tr>
</tbody>
</table>

and two VPL platforms (MBlock and Open Roberta Lab) that use visual blocks instead of text commands.

Comparing the text coding and block-based solutions in Table 2, Table 3 and Table 4, we can first observe that the technical language used by Arduino C is not the most practical for young students, who are not familiar with technical languages. For instance, the students might not understand terms such as “digitalWrite” or “HIGH” or “LOW”\(^{15}\). The same issue arises with the MBlock editor, which, although it uses visual blocks to change the text commands, still uses the same terminology.

In contrast to Arduino C and MBlock, Open Roberta Lab uses more widely recognised terms (“ON-OFF”, for example, instead of “HIGH-LOW”) and, in general, uses terminology that makes more sense for children. For instance, instead of “digitalWrite” or “set digital pin”, it speaks a more human language: “turn LED on” or “play buzzer” with particular “frequency” and “duration”.

These examples show that, if we wish to provide learners with a more intuitive and meaningful programming interface, it is not enough to move from text coding to visual blocks; we need, at the same time, to rethink the role of a technical language that is well established with professionals and hobbyists but may be strange for young learners outside these communities. This reasoning applies not only to text coding but also to the terminology used inside the visual blocks. The aim should be to use a more “human” language that is meaningful for young learners.
Figure 1. Making a simple LED circuit with Chibi Chip board and strips of copper tape as conductive material.

Figure 2. The lighthouse project with Chibi Chip board, an LED circuit and some crafting.

Figure 3. Simulation of the Chibi Chip board (left) and the code for the lighthouse blinking with MakeCode (right).
Table 1 shows that different technologies come with different advantages and shortcomings. For instance, we have noted that LEGO Mindstorms abstracts unnecessary details, limits the time spent on technicalities and is familiar to children; however, it also has only a low potential of becoming a design material and carries a high cost. On the other hand, Arduino offers, at low cost, a high potential of becoming a design material, but it does not abstract unnecessary details and requires considerable attention to technicalities.

With these remarks in mind, we welcome efforts to combine different technologies, taking the strengths of each kit and not its weaknesses. For instance, imagine a solution that would combine the abstraction of LEGO Mindstorms with the power and flexibility of Raspberry Pi or Arduino. Indeed, such solutions have already been proposed and entered the market. For instance, the BrickPi board and the PiStorms board, among others, replace the LEGO NXT/EV3 brick with the Raspberry Pi or Arduino board and offer an extension that makes it compatible with the LEGO Mindstorms platform. For this purpose, a “shield” is added to the Raspberry Pi or Arduino board that offers connectors to the LEGO Mindstorms sensors and motors. Combining the processing power of the low-cost Raspberry Pi or Arduino board with the convenience of the LEGO motors, sensors and building system results in a better tool than either system offers alone.

In another example, the well-known littleBits series offers the littleBits Arduino Bit, a microcontroller that gives the power to code the littleBits circuits and control the way lights, buttons, motors, sensors and other Bits interact. The Arduino Bit can offer the functionality of an Arduino Leonardo without any breadboarding, soldering or wiring. It is simply plugged into a computer, snapped together with other Little Bits and programmed with the visual blocks–based Code Kit App or with MakeBlock (Figure 4).

Very often, a breadboard is used in connection with a microcontroller to make electric circuitry. Our experiences in our pilots with young learners have shown that difficulties and misunderstandings arise for these students while working with a breadboard; they do not understand how its holes are connected internally or why the breadboard differs from the standard diagrams used in physics lessons. Can we improve the design of the breadboard to make it friendlier and more meaningful for young learners?

In an interesting project aimed at strengthening the understanding of the breadboard and closing the gap between its representation in the diagram and the physical breadboard, a 3D-printed plastic cover was designed as an overlay to fit on top of a standard breadboard. The cover visualised the internal connections between the holes, and the supply lines were coloured red and black. In addition, a new type of diagram was designed to complement the new type of breadboard and to further explain the components and connections in comparison with a standard electrical diagram. In the eCraft2Learn pilots, we found that working with a simulation of the breadboard in Tinkercad Circuits software before going to a real breadboard helped young learners understand how to use the breadboard in real circuits.

Several efforts and solutions have focused on adapting the popular Arduino-like microcontrollers to make them friendlier and more manageable for young learners, especially novices in electronics and coding. One popular solution is the use of a “shield”: a piece of hardware that can be mounted on a microcontroller to give it a specific purpose or extra capabilities. For example, a motor shield makes it easier to control motors with a microcontroller, and an Ethernet shield connects it to the Internet.

There are many different shields for a wide variety of purposes. The Arduino Education Shield is a custom-made shield designed by Arduino Education, specially tailored for educational purposes to enable quick and easy learning while building projects and to make connecting and prototyping easier through a simplified design. It connects push button...
modules, light sensor modules, power LED modules and more to an Arduino board and extends its capabilities, making the creation of projects easier.

Using shields instead of the corresponding circuitry and wiring to the microcontroller board offers many advantages. Learners do not need to worry about the circuitry since all the components they need are on the shield. They can easily detach the shield from the microcontroller board and reattach it whenever they want, without worrying about making the circuit and wiring everything again. Using shields leads to fewer errors than the process of connecting the separated parts. Finally, shields offer an easy way to add new features to an Arduino-like board that otherwise would be difficult to create.

Grove by Seeed Studio is another interesting solution that provides an open, modular system designed for easy connection of sensors or actuators to a microprocessor, thus making it easy to connect, experiment and simplify the prototyping process without wiring or soldering. Recently, Arduino and Seeed Studio announced the Arduino Sensor Kit – Base: a plug and play addition to the Arduino board to get users started with electronics and programming. The kit helps to connect and program basic Grove modules that include both sensors and actuators – again without requiring a breadboard, soldering or wiring. Children can also use the compact and flexible Proto Shield kit to build their own Arduino Shield. The Proto Shield kit makes it easy to design custom circuits and solder electronics directly on it.

A simplified type of shield is Snapino, which is helpful for beginners and novices and offers a useful introduction to the Arduino board. Snap Circuits use electronic blocks that snap onto a clear plastic grid to build different circuits. The Snapino module is an Arduino UNO microcontroller mounted on a Snap Circuits base. When combined with Snap Circuits, which has electronic parts and modules mounted on snaps, it offers a simple and easy prototyping platform. Because LEDs come with internal resistors, users do not need to worry about protecting them. A base grid is provided instead of a breadboard for mounting parts and wires.

To conclude the discussion, we present a common project in educational robotics using some of the tools introduced above to show how these can be combined to help young makers create easy, meaningful and realistic projects. The hypothetical project involves designing a robotic car. The task is to design it so that when it receives a signal from a sensor (as a result, for example, of a button being pressed) its lights turn on and it moves forward. In Figure 5, the robot configuration is illustrated as it was made in the Open Roberta Lab software. In Figure 6, the real design is demonstrated; a Grove shield is attached to an Arduino board where a button, an LED and a step motor driver module are snapped without any soldering, breadboard or wiring.

Figure 7 illustrates the code required to control the robotic car in Open Roberta Lab; coloured blocks are easily snapped together, preventing syntax errors. The visual form of the blocks (e.g. repeat indefinitely) aids understanding of the underlying concepts (e.g. loop). The terms in the blocks are remarkably close to normal usage, making the programming task meaningful for young learners. The blocks can easily be parameterised (e.g. rotations per minute, number of rotations) in a flexible manner.

![Figure 5. Robot configuration in Open Roberta Lab.](image-url)
Conclusions

This paper explores technological solutions and tools that can contribute to a more inclusive robotics education. We review key dilemmas regarding the use of technologies in robotics education, offering a perspective that might be useful in both evaluating technologies currently in use and developing new tools and solutions to promote an inclusive robotics education. We criticise deficiencies and shortcomings of technologies currently popular in the field and emphasise that technologies brought into classrooms should be those designed for learners and not for professionals or hobbyists.

Technologies designed for professionals are usually difficult for young learners to understand and operate, and we must carefully consider their use in education in order to prevent a technocentric approach in which learning is focused only on how to use the tool. Such a technocentric approach is precisely the opposite of the inclusive approach we adopt in this paper. We suggest several technological solutions and present an indicative scenario that aims at a didactic transformation of the “difficult” technologies before putting them in the hands of young learners. This didactic transformation involves the transformation of professional or scientific tools in a way that optimises comprehension and ease of use in the educational process.

Finally, this paper highlights the educational value of the “arts and crafts” model and the need for robotics kits that can
become a hands-on design material for children to manipulate and think with. The paper provides examples of the kind of inventive and creative technologies we consider essential to an inclusive educational paradigm.

Further work is required to understand in more depth how technologies determine the models currently in use in robotics education and how designing new technologies or devising new ways to use existing technologies might help educational robotics communities to move away from technocentric models and towards an inclusive educational paradigm.

Data and Software availability
No data are associated with this article.

References

The paper presents an evaluation of popular robotics tools available for educators to use in the classroom as part of a larger project called "INBOTS: Inclusive Robotics for a Better Society". The evaluation focused on how the tool can promote learning opportunities for all students inclusively. The author came up with a set of eight criteria to assess the inclusiveness of those tools evaluated. The platforms evaluated were Lego Mindstorms, Arduino, and Micro:bit. The author extended the evaluation to include and assess other tools including programming environments which can be used with the three platforms to provide a more holistic experience with those tools.

The author uses the term "new paradigm" to explain the curriculum that integrates Papert's constructionism and maker movements while focusing on creativity and 21st-century skills. However, those have already been used in educational contexts for more than 10 years. If this approach is new in Greece, it needs to be addressed in the paper.

Regarding the evaluation of the robotics tools, there is no explanation of why those three tools were selected. There are more and more robotics tools becoming available every year. Although those are some of the most popular tools available, there are other tools similar to those three selected. For example, VEX IQ, Robotis' OLLO, Sony's KOOV are some of the robotics tools similar to LEGO Mindstorms.

Although the evaluation is comprehensive and touched upon details, some of the categories used to evaluate the tools are more instructional problems than tool-specific issues. For example, the first category of DIY robotics can be influenced by instructions/lessons. For example, craft materials can be used with LEGO Mindstorms to create maker/DIY projects. Although Mindstorms is too big to use with a wearable type of project, it can create dancing robots or robotics toys with which costumes are created using craft materials. Arduino can be used with a project for self-expression using various shields and kits, such as the ones that are mentioned in the paper. Again, it is more instructional than tool-specific.

Although each robotics tool has some limitations, which tool to use should be guided by the
learning goals and outcomes of a lesson/curriculum. If the learning objective is to introduce an electric circuit, Arduino is better than Mindstorm.

Another factor that should be taken into account when deciding on learning tools to use is the age/developmental stage of students. In the paper, it is not clear which age group the evaluation has targeted. The evaluation of the tools should include the developmental appropriateness of each tool. Although Arduino is a good tool for DIY type projects, it is not developmentally appropriate for lower elementary school students. Although the paper contains useful information for classroom teachers and educators considering using robotics tool to be integrated into their teaching practices especially focusing on inclusive robotics, instead of starting the paper with the information about how each technology satisfies the criteria emerged from the INBOT project and introducing more options that could make the tool more effective tool in terms of satisfying the criteria, I would suggest organizing the discussion around the criteria and how to make each tool effective to integrate the various pedagogies and approaches INBOT promotes for teachers/educators to use in their classrooms.

Lastly, additional English edit of the manuscript will be helpful.

Is the topic of the review discussed comprehensively in the context of the current literature?
Partly

Are all factual statements correct and adequately supported by citations?
Partly

Is the review written in accessible language?
Yes

Are the conclusions drawn appropriate in the context of the current research literature?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Educational robotics, STEM education, Computer Science Education, curriculum development, AI literacy.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 10 Jun 2021

Dimitris ALIMISIS, EDUMOTIVA (European Lab for Educational Technology), Kapnikareas 19A, Athens, Greece

Thanks for your time to review this paper and for your constructive comments and suggestions. Below are my responses to your comments/suggestions and the revisions
made in the paper.

Regarding the term "new paradigm": I agree with you that constructionism and maker movement have already been used in educational contexts, however the dominant framework seems to be different when it comes to the several curricula provided by the producers of popular commercial robotic kits which certainly influence the use of the kits by teachers and children and robotics education inevitably. A recent review [1] of some typical educational robotics curricula supporting well-known technologies has revealed a chaotic landscape where curricula rarely share common elements and principles. Many of them propose usually a restricted step-by-step guided approach for learners to assembly and program prescribed models of robots. This approach is far away from a constructionist or maker education framework and cannot foster creativity, inventiveness, and curiosity for children [1]. We have criticised somewhere else [2] the deficiencies of the current technocentric [3] paradigm in robotics education and have argued for the need of a paradigm shift [2]. This requires a radical change in the current core concepts and practices of robotics education with the focus being on the transformation of robotics education towards an education inclusive of all children which in this sense represents a new paradigm. I have now explained this issue in two new paragraphs added in page 2 & 3.

Regarding the selection of the reviewed tools, I have now revised the 1st paragraph in p. 6 to better explain why those three tools were selected. In short, the three tools are among the most popular, represent quite different approaches within educational robotics and have attracted much attention by the research community.

“some of the categories used to evaluate the tools are more instructional problems than toolspecific issues”: I agree that some of the categories used to evaluate the three platforms can be influenced at some extent by the teacher’s instructions. Indeed, an aware teacher may suggest some crafting with Lego Mindstorms to decorate the robot but not to make a robot from scratch. And yes, Arduino, if combined with shields or kits like those I recommend in the paper, can improve its potential for self-expression. However, according to a recent review [1] and our experiences from schools at EU level these practices are not the usual or dominant case in schools and the evaluations in Table 1 are based on the currently dominant use of the tools, not the desired ones. While some of the achievements of a certain tool might be improved at some extent if coupled with an aware teacher and a certain pedagogy, I believe that the philosophy and architecture of each tool usually implies (or promotes) a certain learning pathway for users with consequent benefits or shortcomings like those depicted in Table 1. I have now added a paragraph in page 7 to address this issue.

Regarding the age appropriateness of the reviewed tools, I have now added in page 6 age groups appropriate for each reviewed tool.

Regarding the organisation of the discussion: The current structure of the discussion is actually organised around the criteria set; it starts with the evaluation of the three popular platforms to explore how well they satisfy the criteria set which is a necessary step before going to suggestions for alternative or improved technologies made in the final part of the discussion and based again on the criteria set (pages 8-11).


**Competing Interests:** No competing interests were disclosed.

**Reviewer Report 11 May 2021**

https://doi.org/10.21956/openreseurope.14391.r26784

© 2021 Matthews S. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Sarah Matthews**  
Faculty of Engineering, Architecture and Information Technology, The University of Queensland, Brisbane, Australia

This well-written review addresses a set of questions surrounding commonly-used classroom technologies, namely Mindstorms Lego, Arduino and Micro:bit, as well as programming environments for robotics and other maker platforms. It comparatively discusses existing technologies for their ability to support widescale ‘inclusive’ robotics education in schools. The author’s analysis identifies a set of criteria that should be considered when focusing on the potential of hardware and software to be a ‘creative material’ for students, and to support the development of skills such as critical thinking and problem-solving. This is a clear strength of the paper. The discussion is framed well with respect to the different agendas for which technologies are brought into the classroom e.g., for the learning of ICT, the application of engineering concepts, or to encourage meta-cognitive skills such as problem solving or creativity. The paper also introduces the INBOTS project's results to date and proposes a hypothetical educational robotics project that makes use of existing technologies to enable children to easily make creative and meaningful interactive robots.

The topic is timely—insights regarding the nature of and possibilities for technologies to become more than a new content area in educational curricula is a key current issue. This is evidenced by contributions to the recent special issue of International Journal of Child-Computer Interaction (Vol. 29, 2021) on *Computing, Design and Making in Education*. 
Below are recommendations to strengthen and clarify the initial version of the paper.

The paper employs an important but potentially idiosyncratic concept of inclusiveness, as one that “benefits all children regardless of their future educational or professional orientation”; i.e. ‘inclusive’ appears to refer to treating technology education as a general, rather than a specialist, competency, thereby including all students. This is distinct to more orthodox uses that tie inclusivity to accessibility for children with disabilities, e.g., Bar-El et al. (2021)1. An explicit definition of inclusive, or clarification of this distinction for readers might avoid some potential misreadings of the motivation of the project.

Readers who are looking to apply the criteria on page 4 would benefit from knowing from where each of the questions originated. While most of the questions have motivated or have been the subject of previous studies, two of the criteria are introduced without reference to prior work. An explanatory note detailing their provenance, even if they derive from (non-citable) sources such as personal experience or anecdote, will both add clarity and help readers determine the degree of importance to attach to the individual criteria.

Table 1 presents an important contribution, providing an overview of existing technologies set against the criteria. But it treats each of the criteria as binary yes/no; in doing so it overstates the case for/against certain technologies. A qualitative, or possibly just a +/- scale would seem to be more accurate, particularly for the more qualitative criteria.

The recommendation on page 6 to “rethink the role of a technical language... [that] may be strange for young learners”... and to use a more “human” language in coding is one that some other researchers in this space would debate. It might be worth the paper noting in passing, (with Papert), that while it is important to lower the floor to participation, it is also important to do so in ways that do not also lower the ceiling for what children can learn and build with them (Papert, 1980)2. This appears to be where some of the resistance to less technical, more “human” programming languages stems.

Some additional references to existing research and the issues the paper has identified would better ground the selection of these particular technologies for explicit review: Lego Mindstorms: (Castledine, 2011)3 (Üçgül, 2013)4 raise issues that are involved in implementing Mindstorms in a problem solving environment; Arduino: (Agatolio and Moro, 20175; Bekker et al., 20156; Smith et al., 20157); Micro:bit: (Sentance et al., 20178; Videnovik et al., 20189).

The structure of the “Discussion” is not currently visible in the paper. But there are clearly different sections, and distinct purposes that ought to be structurally differentiated: Establishing criteria, Analysing three technologies against the criteria, Reviewing programming languages and tools, Reviewing maker kits, Reviewing friendly adaptations to Arduino, Proposing a hypothetical project. There is currently not a summary of the review(s) to argue how they collectively answer the second research question; it is answered only implicitly in the positive through a proposal for a hypothetical project. A short summary following the reviews that returns to the criteria would be valuable.

References
1. Bar-El D, Worsley M: Making the maker movement more inclusive: Lessons learned from a


Is the topic of the review discussed comprehensively in the context of the current literature?

Partly

Are all factual statements correct and adequately supported by citations?

Partly

Is the review written in accessible language?

Yes

Are the conclusions drawn appropriate in the context of the current research literature?

Yes

**Competing Interests:** No competing interests were disclosed.

**Reviewer Expertise:** Interaction design, educational technologies, creative materials for children, analysis of children's interactions with technologies.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.
Thanks for your time to review this paper and for your constructive comments and recommendations. Below are my responses to your comments/recommendations and the corresponding revisions made in the paper.

Regarding the concept of inclusiveness: this paper is based on the framework of the INBOTS project (Inclusive Robotics for a Better Society), and more specifically on WP3 (Promoting Highly Accessible and Multidisciplinary Robotics Education Programs). The concept of inclusiveness in INBOTS relates to the broader social uptake of robotics. In line with this, inclusive robotics education in the paper refers to an education that will make the multiple benefits coming from robotics education accessible for all the children, demystify robots and familiarise children with them, develop skills for all, and finally promote the future adoption of robots in their work or everyday life. This concept of inclusiveness comes to challenge the misconception that fluency with robotic technologies is one more vocational skill relevant only for students who will follow robotics or engineering careers in the future. This misconception, coupled with gender-biased views that robotics is a “male” subject, may result in discouragement or exclusion of students not interested in engineering careers and especially the female ones. Hence, the concept of inclusiveness in this paper means a robotics education that offers knowledge and skills valuable for every citizen and is clearly distinct to accessibility for children with disabilities. I have now added a paragraph in page 2 to clarify the concept of inclusiveness.

“two of the criteria are introduced without reference to prior work...” I have now added references to literature and to our experiences from a previous project to help readers know the provenance of the questions (Should technologies prepare students for engineering jobs or develop personal expression and skills? Cost matters) (in page 5 & 6).

Regarding your suggestion for a qualitative scale in Table 1, I make now use of an ordinal scale consisting of a spectrum of qualitative values for comparing the performance of platforms (excellent, good, average, poor, very poor).

Regarding your comment on the “low floor – high ceiling“ issue in programming tools: “Technical language“ and “human language“ in the paper refer to the terminology used not only in text coding (e.g. Arduino IDE) but also inside the visual blocks in many programming tools intended for young learners which in many cases just replicate the terminology used in text coding intended for professionals. Our main concern in this paper is how to make programming work meaningful and accessible for young learners removing barriers imposed by an unnecessarily technical terminology; hence, the focus in this paper is to ensure a low floor for young learners in programming. While I believe the recommendation for making programming meaningful for learners should be considered also when it comes to “high ceiling”, I agree with you that efforts to provide a low floor in programming should not undermine a high ceiling (which is important for learners to reach). I find useful to this end the opportunity for young learners to have parallel access to visual blocks and text
coding tools. For instance, when learners program their Arduino board using visual blocks in Open Roberta Lab, Arduino C text coding is automatically generated and appears in a second window. This parallel access helps to familiarise children with C coding and prepare them to jump into a more advanced programming tool when they feel ready to go. I have now added in p. 8 a paragraph addressing your comment.

“Some additional references to existing research and the issues the paper has identified” have now been added.

“The structure of the “Discussion” is not currently visible in the paper”: Some subheadings have now been inserted to make the structure visible.

“A short summary following the reviews that returns to the criteria would be valuable”: I have now added in page 11 a summary of the reviews.

**Competing Interests:** No competing interests were disclosed.